強化学習による理論解析手法の開拓
~Alpha Zero for Physicsに向けて~

Yoshihiro Michishita(Riken CEMS / Proxima Technology)

（コードはhttps://github.com/YoshihiroMichishita/julia/tree/master/AlphaZeroForPhysics で公開中）

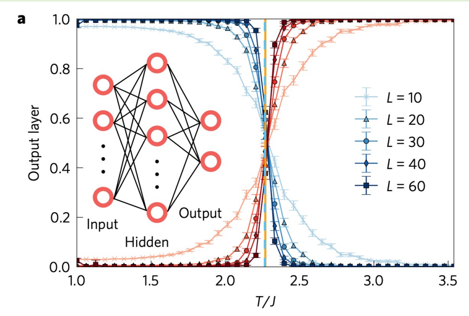
(part1の内容は、https://github.com/YoshihiroMichishita/julia/blob/master/tutorial_RL.ipynb をどうぞ)

Outline

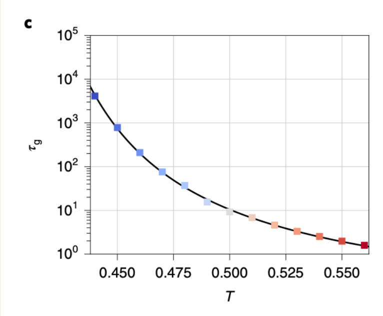
- **Introduction**
  - 機械学習と物理学
  - 物理学における理論解析手法
  - 今回の研究の目標
- **Quick Review**
  - 強化学習(RL)
  - Alpha Zero
  - Symbolics Physics Learner
- **Results**

Recent application of ML to physics(＊物性屋目線です)1/14

Detect the phase transition



Ising magnetization
(Nat. Phys: 13.431(2017))



glass transition
(Nat. Phys: 10.1038(2020))

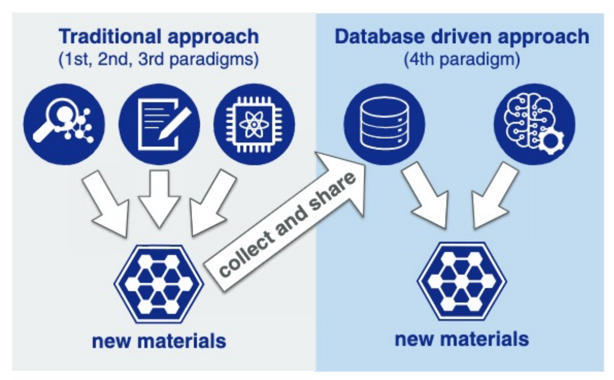
Calcu. equilibrium or steady state

RBMでスピン系の計算(Science: 335.602-606(2017))
(PRB: 96.205152(2017))

RBMでSSの計算 (PRB: 99.214306(2019))

RNNを用いた基底状態の表現(PRR: 2.023358(2020))

Materials Infomatics



(Advanced Science: 6.1900808(2019))

Remove noise or enhance the accuracy

Gaussian Processを仮定して (IEICE: 10.1587(2010))

NV centerに適用 (Sci. Rep.: 12.13942(2022))

理論解析手法とは何か？

●Scale separation & Reduction

- Nonlinear system => reduction
- The Hubbard model (Lattice model)=> Heisenberg model
- Open quantum system => Markov app., GKSL equation
- Periodic driving system => high-frequency expansion
- (Renormalization Group => cutoff scale)
- (DMRG, Tensor network => SVD & reduction)

適切な射影orユニタリ変換が必要

＊下の二つも人為的なスケール分離(カットオフ)を導入して次元削減を行うので、
「scale separation & reduction」に分類される

理論解析手法とは何か？

- 言いたい事
  - スケールの分離がある場合、変数の消去(redcution)が出来る場合がある
  - スケールの分離がある時に、reductionや摂動論を用いたい場合、一般にはそれが出来るframeにユニタリ変換や射影を行う必要がある

なので、物理の(手でできる範囲の)理論解析とは、

ここを機械学習(強化学習)にやらせたい

1. (scaleの分離がある場合に)摂動論やreductionが出来る良いframeを見つける
2. (妥当な近似として)摂動論やreductionを実行し有効模型を作る
3. 個々の物性を得られた模型で解析

強化学習のすすめ（個人的なお気持ち）

●理論からのアプローチとして(良質な)学習データは集めづらい

- 既存の数値シュミレーション手法でデータを集める=>既存の手法の置き換えにしかならない
- 既存の手法でアプローチできない領域で正しさを確認出来ない
- 計算資源勝負みたいなところもあるのでアカデミアで研究するのはしんどい(?)

●強化学習は事前データが要らない

- どちらかと言えば探索アルゴリズム大事(?)
- 解析手法の探索はこれしかない(？)
- 物理で有用なフレームの探索はターン制のゲームだと思うとAlpha Zero等のコードが流用できる

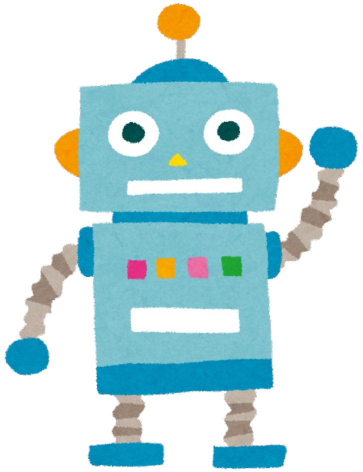
Quick review of RL

- **RL**

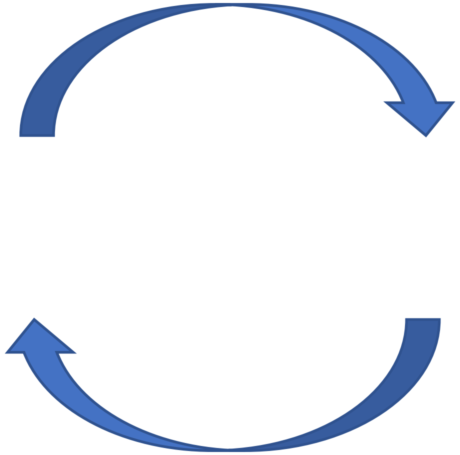
・環境・エージェントから成る系を考える
・環境の状態を与えられ、エージェントが判断して行動を起こす
・エージェントの行動によって環境が変化(行動によって報酬が与えられる)

エージェント

探索と活用



行動: $a_t \in \mathcal{A}$



環境

方策関数: $\pi\ (s_t \to a_t)$

状態: $s_t \in \mathcal{S}$,報酬: r_t

行動価値関数: $\mathrm{Q}\ (s_t, a_t)$

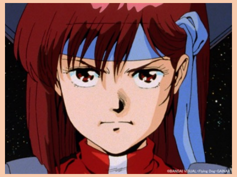
(例えば)$\sum_t r_t$を最大化するように方策をアップデート

Alpha Zero

●概要(ざっくりしたイメージ)

・プレイヤーとコーチ(NN)と環境(ボードゲームのルール・盤面)からなる

プレイヤー



(P + UCT (MCTS)による探索)

探索の結果をコーチに伝える

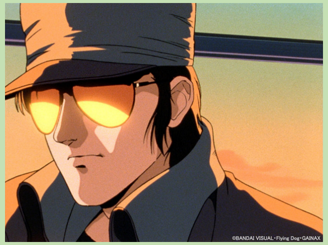
盤面の状況
ルール

コーチのアドバイスを
ある程度従い探索

アドバイスと
行動の評価値を教える

環境

コーチ



(CNN+ Dual(Dueling) network)

探索の結果を踏まえて
コーチングを修正(学習)

●プレイヤーについて

➤ UCT(Upper Confidential Bound (applied to Trees))

$$a_t = \mathrm{argmax}_{\{a \in A\}}[q_t(s_t,a) + c\sqrt{\frac{\log(\sum_a m_t(s_t,a)+1)}{m_t(s_t,a)+1}}]$$

(UCB… リグレット(探索の効率の悪さ)の上限をO(log(t))で与える探索法)

コーチの行動評価　やった事ない行動を時々試してみる

シミュレーションでq値と訪問回数mの更新を繰り返しながら上記の式でシミュレーション行動を選択、
一定数回シミュレーションを行った後、最も練習した(シミュレーションした)行動を選択(本番行動)。
ゲームが終了するまで行動を実行。これを繰り返す。
(練習の後に本番。本番の成果と練習内容をコーチに伝える)

➤ P+UCT

$$a_t = \mathrm{argmax}_{\{a \in A\}}[q_t(s_t,a) + c\, p(s_t,a)\sqrt{\frac{\log(\sum_a m_t(s_t,a))}{m_t(s_t,a)}}]$$

コーチからのオススメ行動(深く探索するために重要)

Alpha Zero

●コーチ(NNの中身, 学習)について

➢ **中身**　　(CNN+ Dual(Dueling) network)

(3*3の畳み込み層+Batch正則化+ReLu+ResNet)*19 (ここで状況判断)

1*1の畳み込み層+Batch正則化+ReLu
(次の一手の優先度)(コーチのアドバイス)

Batch正則化+ReLu + tanh
(勝率)(コーチからの行動の評価)

➢ **学習**

$$L(m) = \frac{1}{N}\sum_{n}\left(r_n - v(m(s_n))\right)^2 - \pi_n \log p(m(s_n)) + \eta \sum m.w^2$$

コーチの行動評価の間違い

コーチのアドバイスを聴いてくれたか(有効さ)

過学習しないためのweight decay
(過激なコーチングの禁止)

Symbolic Physics Learner

●物理の方程式を木の形でする=>AlphaZeroと同じ文脈に載せられる

Published as a conference paper at ICLR 2023

SYMBOLIC PHYSICS LEARNER: DISCOVERING GOVERNING EQUATIONS VIA MONTE CARLO TREE SEARCH

Fangzheng Sun
Northeastern University
Boston, MA, USA
sun.fa@northeastern.edu

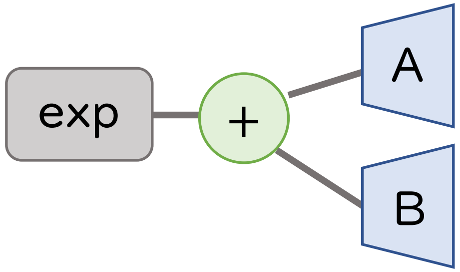
Yang Liu
University of Chinese Academy of Sciences
Beijing, China
liuyang22@ucas.ac.cn

Jian-Xun Wang
University of Notre Dame
Notre Dame, IN, USA
jwang33@nd.edu

Hao Sun*
Renmin University of China
Beijing, China
haosun@ruc.edu.cn

与えられたダイナミクスに近い方程式を木で表現して
MCTSで探索(論文では二重振り子のダイナミクスの推定)

$\exp[A+B] =$



Nodeの種類としてfunction, branch, variableの３つで表現

例えば縮約や射影を行いやすいようなフレームに写すユニタリ変換が知りたい時に、
肩に乗ってるのはエルミートな演算子なので

Function…$\sum_i$, exp, log, $\int dx$, ∂_x　Branch… +, −, −i[,], { , }　Variable…ψ_i, $\psi_i^\dagger$

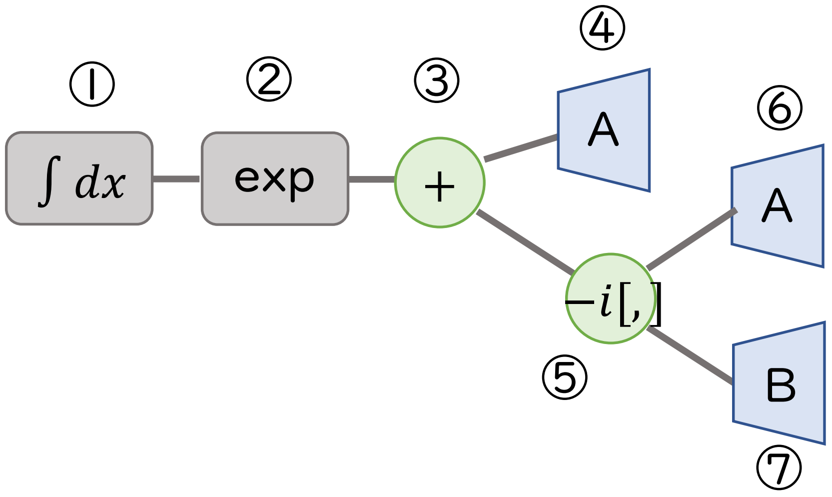
(この場合は、多体系でも手の広さは将棋の中盤と割と同じくらい？)

Symbolic Physics Learner

●Symbolic Physics Learnerにおける“ルール”

・SPLの手順

$$\int dx\, \exp[A - i[A, B(x)]] =$$



・SPLの“ルール”

1. (Variableの数) ≤ (現存するBranchの数+1) になるようにする(等式成立で方程式が完成)

2. Branchの後に同じBranchを続けてはいけない(方程式(演算)の対称性からくるルール)

3. Expの後にlogを連続させない(冗長性を消す)(方程式における“千日手”)

Function…$\sum_i$, exp, log, $\int dx$, ∂_x　Branch… +, -, -i[,], { , }　Variable…ψ_i, $\psi_i^\dagger$

Quick review of Floquet theory & RF

➢ Formalism

$$\hat{H}(t) = \hat{H_0} + \hat{V}(t) \qquad i\frac{d}{dt}|\psi(t)> = \hat{H}(t)|\psi(t)> \qquad \hat{U}(t) = \exp[i\hat{K}(t)]$$

$$i\frac{d}{dt}|\tilde{\psi}(t)> = i\frac{d}{dt}\hat{U}(t)|\psi(t)> = \hat{H}_r(t)|\tilde{\psi}(t)> \qquad \hat{H}_r(t) = \hat{U}(t)(\hat{H}(t) - i\partial_t)\hat{U}^\dagger(t)$$

$\hat{H}(t) = \hat{H}(t+T)$のとき、$\hat{U}_F(t) = \hat{U}_F(t+T), \hat{H}_r(t) = \hat{H}_F$ を満たす$U_F(t)$が存在

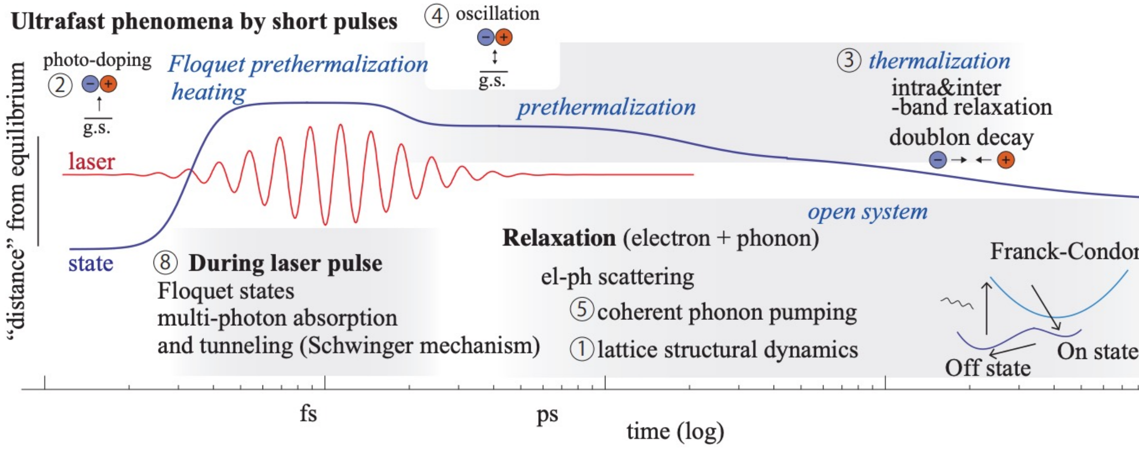
さらに高周波領域$||H_0|| \ll \Omega$の時、$U_F(t), H_F$を$1/\Omega$展開する方法がある(van-Vleck, Floquet-Magnus)

$$H_r(t) = H_F^{(n)} + O(\frac{1}{\Omega^{n+1}}, t)$$

➢ Floquet pre-thermalization

Hamiltonianがlocalかつ、$t = mT$の時
(arXiv:1509.03968(2015))

$$||\mathcal{T}\exp[-i\int ds\, H(s)] - \exp\left[-iH_F^{(n)}t\right]|| \leqq \exp[-O(\Omega)]t$$



Demonstration

●Model

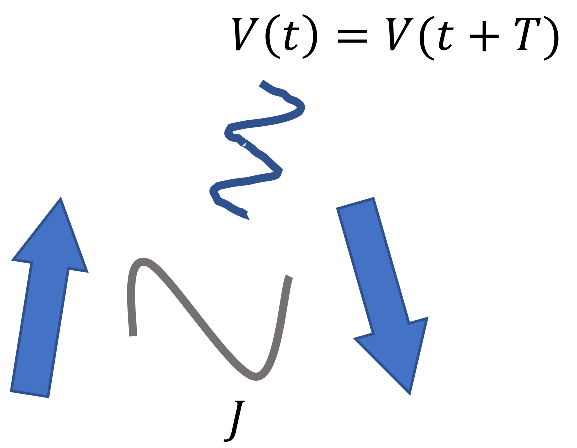
➢ Interacting quantum Two-Spin model under driving

$$\hat{H}(t) = \widehat{H_0} + \hat{V}(t)$$

$$\widehat{H_0} = -\sum_{\alpha}(J_\alpha \widehat{S_1^\alpha} \otimes \widehat{S_2^\alpha} + h_\alpha \sum_i \widehat{S_1^\alpha})$$

$$\hat{V}(t) = -\sum_{\alpha} \xi_\alpha sin(\Omega t) \sum_i \widehat{S_i^\alpha}$$

$$\vec{J} = (J_x, J_y = 0, J_z), \qquad \vec{h} = (h_x = 0, h_y = 0, h_z), \vec{\xi} = (\xi, 0, 0)$$

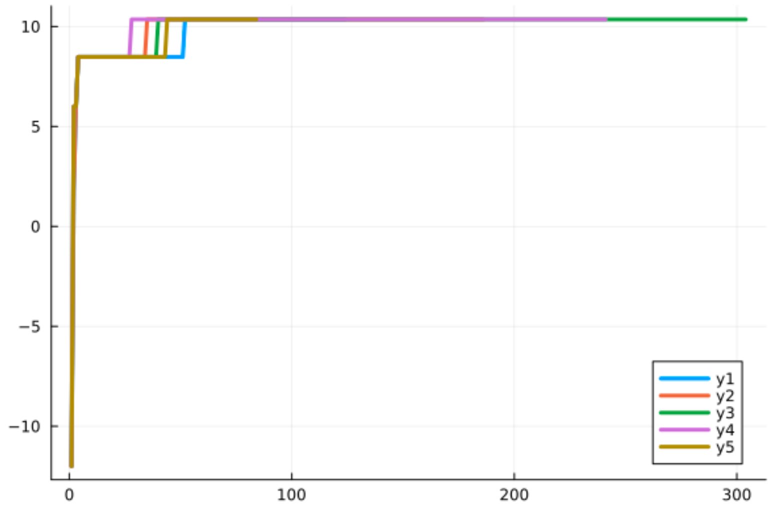


Maximize the Reward = $-log\int dt\, tr(\widehat{H_r(t) - \hat{H}_r(t-\delta t)})$^2

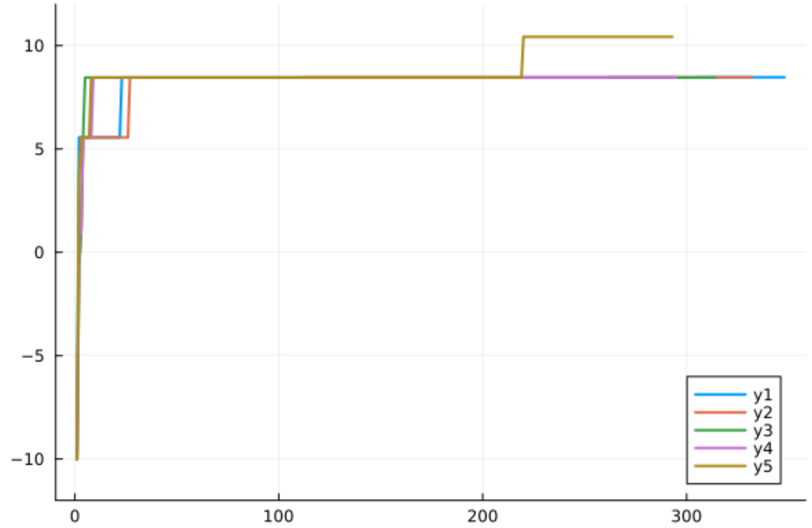
Results

- **Max_length=8の時**

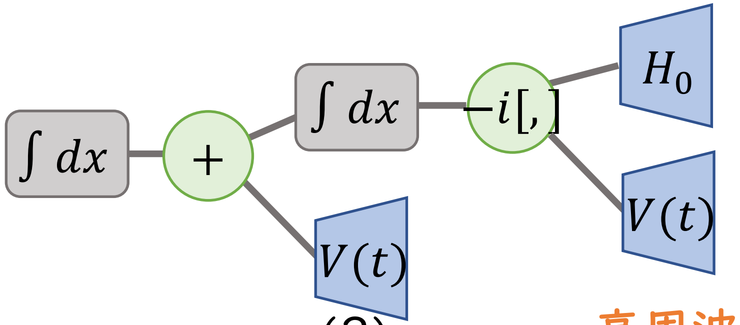
AlphaZero
(ちょこちょこ手を加えてます)



ε-greedy法(強化学習の原始的手法)
(こっちもちょっと工夫しちゃってます)




```
==============
it=2;
##########341.462362 seconds (578.74 M allocations: 21.722
ime)
val=0.19563204, pol=0.69593036
val=0.20644784, pol=0.6926276
val=0.17525849, pol=0.6966044
val=0.22328483, pol=0.71402514
val=0.24023719, pol=0.71967083
val=0.2254533, pol=0.72578335
 51.675910 seconds (101.13 M allocations: 164.628 GiB, 11.8
store data
200
max score: 10.355681;   hist: ∱dt + V(t) ∱dt -i[,] H_0 V(t)
==============
```



(2)

高周波展開と等価!

$$= \int dt\,(V(t) - i\int dt\,[H_0, V(t)])$$

Remarks:

Model:

$$\hat{H}(t) = \widehat{H_0} + \hat{V}(t)$$

$$\widehat{H_0} = \sum_\alpha (J_\alpha \widehat{S_1^\alpha} \otimes \widehat{S_2^\alpha} + h_\alpha \sum_i \widehat{S_1^\alpha})$$

$$\hat{V}(t) = -\sum_\alpha \xi_\alpha sin(\Omega t) \sum_i \widehat{S_i^\alpha}$$

Formalism:

$$\hat{H}_r(t) = \hat{U}(t)(\hat{H}(t) - i\partial_t)\hat{U}^+(t)$$

$$\hat{U}(t) = \exp[i\hat{K}(t)]$$

$$\hat{K}'(t) = \frac{d}{dt}\hat{K}(t)$$

環境が返す報酬:

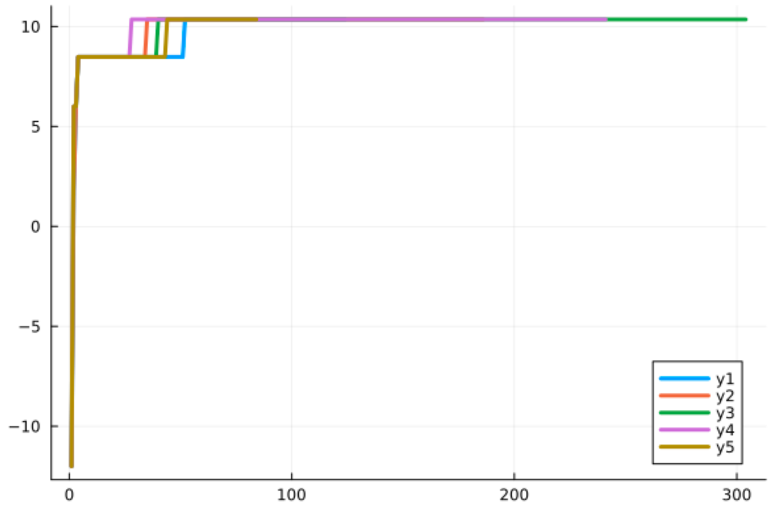
$$-log\int dt\, tr(\widehat{H}_r(t) - \widehat{H}_r(t-\delta t))\hat{}2$$

<u>これを最大化するような木を作る</u>

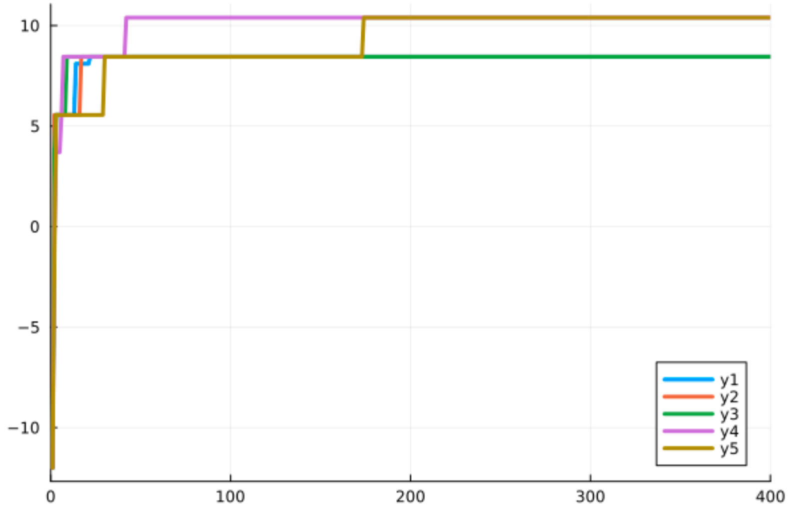
Results

- **Max_length=8の時**

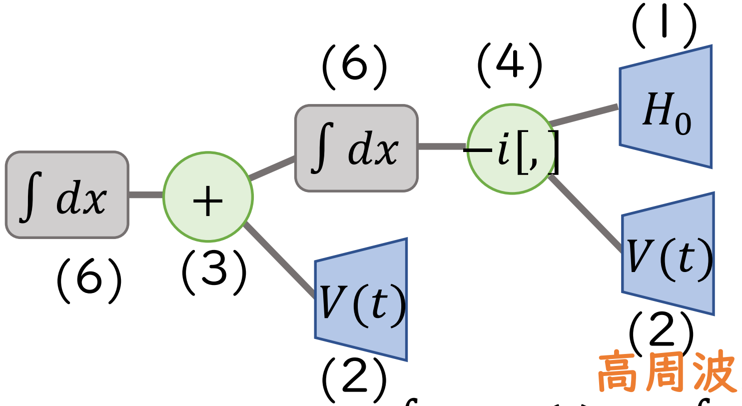
AlphaZero
(ちょこちょこ手を加えてます)



PPO +Actor-Critic法
(Proximal Policy Optimization)




```
it=2;
496.071510 seconds (401.56 k allocations: 134.160 MiB, 0.01% gc time, 0.02%
val=1.5820317, pol=0.61543405
val=1.1977102, pol=0.60332686
val=1.0977467, pol=0.695877
val=1.2222252, pol=0.6558915
val=1.6906229, pol=0.6552366
val=2.8830054, pol=0.69791865
 68.607700 seconds (107.28 M allocations: 28.289 GiB, 4.51% gc time)
store data
435
------------
head = 1;
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
[6, 3, 6, 4, 1, 2, 2], score:10.395253, val(NN):10.398022
```



高周波展開と等価!

$$= \int dt\,(V(t) - i\int dt\,[H_0, V(t)])$$

Remarks:

Model:

$$\hat{H}(t) = \widehat{H_0} + \hat{V}(t)$$

$$\widehat{H_0} = \sum_\alpha (J_\alpha \widehat{S_1^\alpha} \otimes \widehat{S_2^\alpha} + h_\alpha \sum_i \widehat{S_1^\alpha})$$

$$\hat{V}(t) = -\sum_\alpha \xi_\alpha sin(\Omega t) \sum_i \widehat{S_i^\alpha}$$

Formalism:

$$\hat{H}_r(t) = \hat{U}(t)(\hat{H}(t) - i\partial_t)\hat{U}^+(t)$$

$$\hat{U}(t) = \exp[i\hat{K}(t)]$$

$$\hat{K}'(t) = \frac{d}{dt}\hat{K}(t)$$

環境が返す報酬:

$$-log\int dt\, tr(\widehat{H_r}(t) - \hat{H}_r(t-\delta t))\text{^}2$$

これを最大化するような木を作る

Remarks & Outlook

●まとめ

- 方程式を木の形で表現する事で「ある性質を満たす方程式の探索」の問題を「ゲームの最善戦略の探索」の問題にマッピングできる。

  (ので、強化学習で“解ける”)

- Alpha Zeroのアルゴリズムを用いて、「高周波展開」を導出する事ができた。(もちろん他の理論解析手法の導出に対しても使えるはず)

Remarks & Outlook

●Outlook

- いろんな(もう少し難しい)問題に適用してみる
  - ・“環境”部分を変えるだけなので問題を思いつけばすぐ試せるはず
  - ・格子系や力学系への適用
  - ・転移学習が出来るか？

- 「関数ノード」の種類をどう設定するか？
  - ・問題によってはこれが本質的になるので考えないといけない
  - ・「物理の問題でよく出てくる関数セット」みたいなのを考える必要あり