Machine Learning for Tight-Binding Hamiltonian

Project presentation

Ivan SCOLAN[1,2] Paul RUELLOUX[1,2] Alexis WIETZKE[1]

[1]M2 ICFP ENS PSL

[2]Magistère de Physique Fondamentale d'Orsay
Université Paris-Saclay

April 4, 2024

Outline

Brief introduction of the Tight Binding model

$$H = \sum_{j \neq i}^{N} t_{ij} c_i^\dagger c_j + \sum_i \epsilon_i c_i^\dagger c_i$$

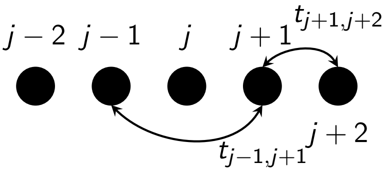


Figure: 1D tight-binding model

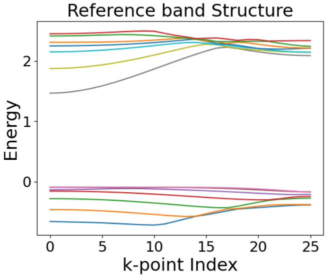
Matrix representation:

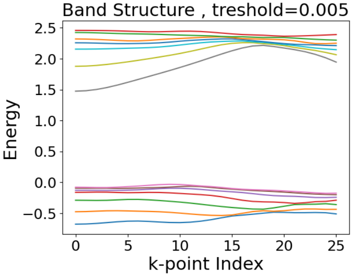
$$H = \begin{pmatrix} \epsilon_1 & t_{1,2} & 0 & \dots & 0 \\ t_{2,1} & \epsilon_2 & \ddots & & \\ 0 & \ddots & & & \\ \vdots & & & & t_{N-1,N} \\ 0 & & & t_{N,N-1} & \epsilon_N \end{pmatrix}$$

Goal of the algorithm

Goal of the algorithm :

- find the values for the t_{ij}'s and ϵ_i's so that the the energy spectrum of the Hamiltonian matches the reference energy bands.
- provide a Tight-Binding Hamiltonian for the system





Class and methods

One class TBHNN() containing six methods

- initialise()
- read_training_set()
- define_TB_representation()
- reset_optimiser()
- reinitialise()
- compute_bands()

Then in our algorithm we have three functions

- main()
- stochastic_reset_fitting()
- no_reset()
- fitting()

Studied parameters

The parameters that we can play with:

- the optimiser (Adam,...)
- the loss (meanLoss, smooth L1 Loss,...)
- learning rate
- amplitude of initialisation σ
- threshold ϵ
- type of resetting (stochastic, fixed, no reset)

In this presentation we will focus on

- type of resetting
- amplitude of initialisation σ

Tuning the reinitialize function

- The reinitialize function
- Hopping terms and energies initialised by random variables following $\sim \mathcal{N}(0, \sigma_{init})$
- Dependency to σ_{init}

Convergence dependency on σ_{init}

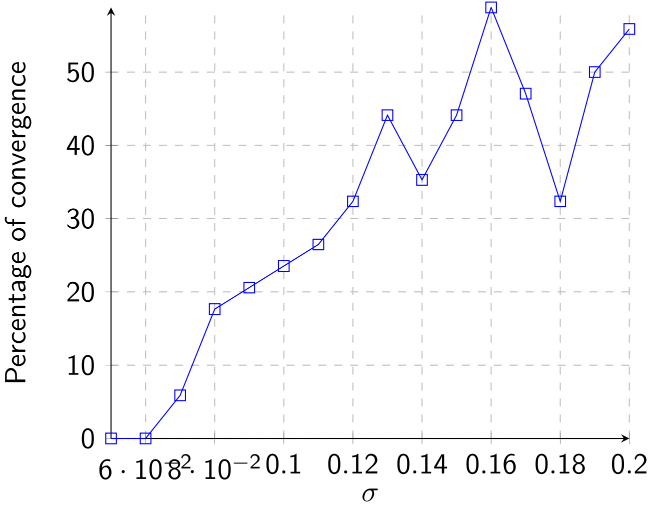


Figure: Percentage of convergence under 8000 steps, depending on σ (threshold$=3e-3$, $\text{nb}_{avg} = 33$).

Average total number of steps for one convergence

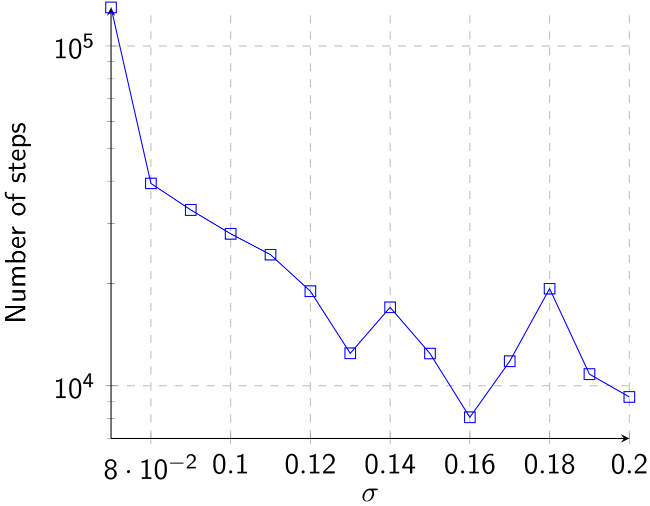


Figure: Average total number of steps for one convergence, depending on σ (threshold$=3e-3$, determinist reset at $d = 8000$, $\text{nb}_{avg} = 33$).

Converging path length dependency on σ_{init}

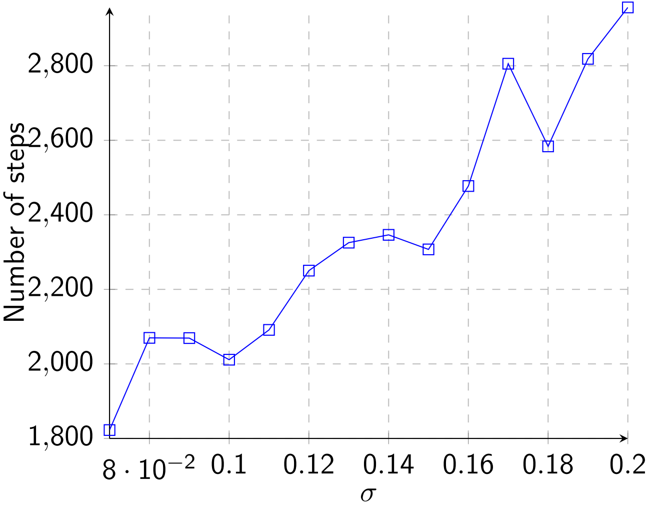


Figure: Average length of a converging run, depending on σ (threshold$=3e-3$, $\mathrm{nb}_{avg} = 33$).

Motivation

- The SGD encounters zero, one or multiple plateau for ϵ small enough
- The energy landscape admits faster paths
- Resetting $\longleftrightarrow$ reinitializing the search to avoid plateau

Implementation

- (D) Deterministic for $\lambda \equiv \#$ steps before resetting
  (S) Stochastic for $\lambda \equiv$ average $\#$ steps before resetting
  & exponentially distributed
- $N \equiv$ total number of steps before convergence
  $k_{S/D} \equiv$ number of resets

$$\Rightarrow k_D \approx \frac{N}{\lambda} \qquad k_S \sim Pois(\lambda N)$$

- N depends on the landscape *i.e.* on the chosen path, which depends on the (re)-initial state

$$\longrightarrow \text{non-trivial } k_s \text{ and } N \text{ dependency}$$

Implementation

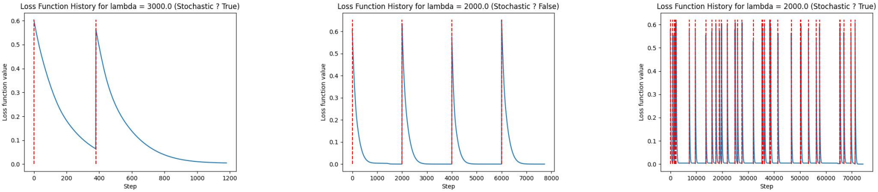


Figure: Loss function history for (S) and (D)

- If ϵ small enough we expect:

$$\lambda \to +\infty\text{: (S) and (D) should coincide: no resetting}$$
$$\lambda < \lambda_{critical}(\epsilon)\text{ : N diverges for (S) and (D)}$$

- Is there an optimal λ at which (S) is better than (D) ?

Comparison of (S) and (D)

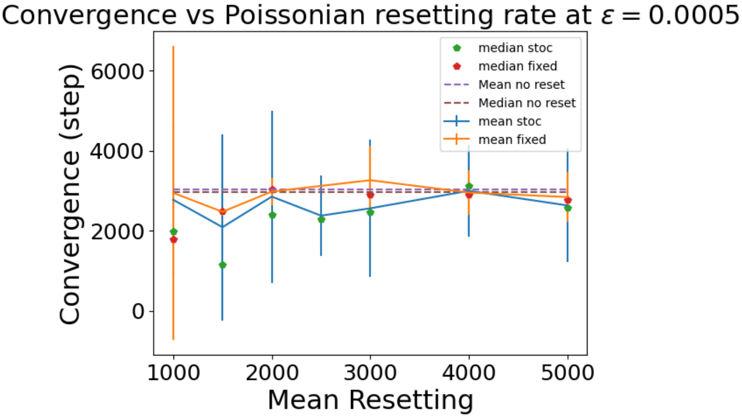


Figure: Comparison of (S) and (D): each point sampled over 20 runs

Stochastic vs deterministic reset

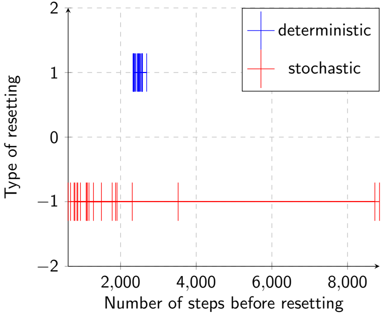


Figure: Comparison of the number of steps between the two resetting methods.

Stochastic Resetting + cutoff in order to avoid rare events
$\Rightarrow$ fastest convergence

Conclusion

- Good reproduction from ab-initio data using SGD
- Dependence on initialisation
- Stochastic resetting is shown to work best on average with a cutoff