



HAL
open science

Some Remarks on Replicated Simulated Annealing

Vincent Gripon, Matthias Löwe, Franck Vermet

► **To cite this version:**

Vincent Gripon, Matthias Löwe, Franck Vermet. Some Remarks on Replicated Simulated Annealing. Journal of Statistical Physics, 2021, 182 (3), 10.1007/s10955-021-02727-z . hal-03238781

HAL Id: hal-03238781

<https://hal.science/hal-03238781>

Submitted on 31 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



Some Remarks on Replicated Simulated Annealing

Vicent Gripon¹ · Matthias Löwe²  · Franck Vermet³

Received: 7 October 2020 / Accepted: 13 February 2021 / Published online: 2 March 2021
© The Author(s) 2021

Abstract

Recently authors have introduced the idea of training discrete weights neural networks using a mix between classical simulated annealing and a replica ansatz known from the statistical physics literature. Among other points, they claim their method is able to find robust configurations. In this paper, we analyze this so called “replicated simulated annealing” algorithm. In particular, we give criteria to guarantee its convergence, and study when it successfully samples from configurations. We also perform experiments using synthetic and real data bases.

Keywords Simulated annealing · Replicas · Perceptron · Neural networks · Optimization

Mathematics Subject Classification 60J10 · 82C32 · 65C05

1 Introduction

In the past few years, there has been a growing interest in finding methods to train discrete weights neural networks. As a matter of fact, when it comes to implementations,

Communicated by Hal Tasaki.

The research of the second author was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy EXC 2044-390685587, Mathematics Münster: Dynamics—Geometry—Structure.

✉ Matthias Löwe
malowe@math.uni-muenster.de

Vicent Gripon
vincent.gripon@imt-atlantique.fr

Franck Vermet
Franck.Vermet@univ-brest.fr

¹ IMT Atlantique UMR CNRS Lab-STICC, Technopole Brest Iroise, 29238 Brest, France

² Fachbereich Mathematik und Informatik, Universität Münster, Einsteinstraße 62, 48149 Münster, Germany

³ Laboratoire de Mathématiques, UMR CNRS 6205, Université de Bretagne Occidentale, 6, Avenue Victor Le Gorgeu, CS 93837, 29238 Brest Cedex 3, France

discrete weights allow to reach a better efficiency, as they considerably simplify the multiply-accumulate operations, with the extreme case where weights become binary and there is no need to perform any multiplication anymore. Unfortunately, training discrete weights neural networks is complex in practice, since it basically boils down to a NP-hard optimization problem. To circumvent this difficulty, many works have introduced techniques that aim at finding reasonable approximations [6,7,13,24].

Among these works, in a recent paper Baldassi et al. [2] discuss the learning process in artificial neural networks with discrete weights and try to explain why these networks work so efficiently. Their approach is based on an analysis of the learning procedure in artificial neural networks. In this process a huge number of connection weights are adjusted using some stochastic optimization algorithm for a given target function. Some of the resulting optima of the target or energy function have better computational performance and generalization properties, other worse. The authors in [2] propose that the better and more robust configurations of weights lie in dense regions with many maxima or minima (depending on the sign) of the target function, while the optimal configurations that are isolated, i.e. far away from the next optimum of the energy function have poor computational performance. They propose a new measure, called the robust ensemble, that suppresses such configurations with bad computational performance. On the other hand, the robust ensemble amplifies the dense regions with many good configurations. In [2] the authors present various algorithms to sample from this robust ensemble, one of them is *Replicated Simulated Annealing* or *Simulated Annealing with Scoping*. This algorithm combines the replica approach from statistical physics with the simulated annealing algorithm that is supposed to find the minima (or maxima) of a target function. The replica technique is used to regularize the highly non-convex target or energy function (another regularization idea was introduced recently in [4]), while the simulated annealing algorithm is used afterwards to minimize this new energy. We will define Replicated Simulated Annealing in Sect. 2.

To give a first impression of this algorithm, assume we have $\Sigma := \{-1, +1\}^N$ as our state space and N is large. On Σ we have very rugged energy function $E : \Sigma \rightarrow \mathbb{R}$ and assume that $\min_{\sigma \in \Sigma} E(\sigma) = 0$. To find the minima of E one could run a Metropolis algorithm for the Gibbs measure at inverse temperature $\beta > 0$:

$$\pi_{\beta}(\sigma) := \frac{\exp(-\beta E(\sigma))}{Z_{\beta}}, \quad \sigma \in \Sigma.$$

Here $Z_{\beta} := \sum_{\sigma'} \exp(-\beta E(\sigma'))$ is the partition function of the model, a normalizing factor that makes π_{β} a probability measure. If one carefully lowers the temperature, i.e. if one increases β slowly enough during the process the corresponding Markov chain will get stuck in one of the maxima of π_{∞} which are easily seen to be the minima of E . This is the classical Simulated Annealing algorithm, cf. [25] or [15] for the seminal papers. The question how to choose the optimal dependence of β from time t and its convergence properties have been extensively discussed. We just mention [5,8,18,20,22], for a short and by far not complete list of references. The upshot is that good a "cooling schedule" is of the form $\beta_t = \frac{1}{m} \log(1 + Ct)$, where m can be roughly described as the largest hill to be climbed to get from an arbitrary state to one of the global minima of the target function.

However, sometimes not all the global minima are equally important, in particular one may be interested in regions with many global minima (or almost global minima), so called dense regions. An obstacle may be, that E exhibits many global minima, but only relatively few of them are in dense regions. Let us motivate this question by a central example which we will often have in mind in this context and which also is one of the central objects in [2].

Example 1 Assume we have patterns $\xi^1, \dots, \xi^M, \xi^\mu \in \{\pm 1\}^N, \mu = 1, \dots, M$ where $M = \alpha N$ for some $\alpha > 0$. Each of these patterns belongs to one of two groups, which we indicate by $\vartheta(\xi^\mu) = \vartheta^\mu \in \{\pm 1\}$.

The task is to classify these patterns. One of the standard methods to do this in machine learning is the perceptron. Perceptron was defined in the 1960s by Rosenblatt [34]. It is one of the first mathematical models of neural networks, inspired by the biological phenomenon of vision. In its simplest form, the one we are studying here, it is a single neuron, corresponding to the mathematical model of Mac Culloch and Pitts [30]. It is then a binary classifier, which separates two sets of points linearly separable by a hyperplane. In mathematical terms it maps its input x to $f(x)$ which is either 1 or 0, and thus puts it into one of two classes. This decision is made with the help of a vector of weights

$$W = (W_1, \dots, W_N) \in \{-1, +1\}^N.$$

More precisely,

$$f(x) = \begin{cases} 1 & \text{if } \langle W, x \rangle > 0, \\ 0 & \text{otherwise} \end{cases}$$

where $\langle W, x \rangle$ is the dot product $\langle W, x \rangle = \sum_i W_i x_i$. These weights $W = (W_1, \dots, W_N) \in \{-1, +1\}^N$ have to be learned and we want the classification to be perfect, i.e. we want that

$$\Theta(\vartheta^\mu \langle W, \xi^\mu \rangle) := \Theta\left(\vartheta^\mu \sum_{i=1}^N W_i \xi_i^\mu\right) = 1$$

for all $\mu = 1, \dots, M$. Here

$$\Theta(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

denotes the Heaviside-function. Hence our classification task is fulfilled if

$$\sum_{\mu=1}^M \Theta(-\vartheta^\mu \langle W, \xi^\mu \rangle) = 0$$

(where we assume that N is even to avoid the specification of tie-breaking rules) or, equivalently

$$\prod_{\mu=1}^M \Theta(\vartheta^\mu \langle W, \xi^\mu \rangle) = 1. \tag{1}$$

Note that in Rosenblatt’s initial model, the weights (W_1, \dots, W_N) , called synaptic weights, are real-valued and not restricted to take their values in $\{-1, +1\}^N$. The objective now is to find weights W such that (1) is true, i.e. we are searching for weights W such that $\bar{E}(W) = -\prod_{\mu=1}^M \Theta(\vartheta^\mu \langle W, \xi^\mu \rangle)$ is minimal. Obviously, this optimization problem is of the above mentioned form. However, one prefers weights W in so called dense regions, i.e. weights that are surrounded by weights \bar{W} that are also minima of \bar{E} . The idea is that these states have good generalization properties or a small generalization error. This means that we want to find weights, that still classify input patterns correctly, which we have not seen in our training set ξ^1, \dots, ξ^M . It is at least plausible that weights with a small generalization error lie in dense regions of $\{\pm 1\}^N$.

In this work we are interested in making explicit convergence properties of the algorithm of Replicated Simulated Annealing. We also perform experiments using synthetic and real data bases. The outline is as follows: in Sect. 2 we mathematically formalize and describe the algorithm of Replicated Simulated Annealing, in Sect. 3 we study its convergence properties. Naturally, this convergence will be studied on an infinite time horizon. This is a slightly different set-up than in [2], where experiments are performed for finite time. However, the question, whether or not Replicated Simulated Annealing converges is the first question that should be analyzed, before studying which choice of parameters yields the best results. This latter question is addressed in Sect. 4, where we perform experiments using synthetic and real data bases. Of course, the time horizon for such experiments is finite. On the other hand, in finite time we can control the influence of the choice of the parameters on the performance which is hard to control theoretically. Finally, Sect. 5 is a conclusion.

2 Replicated Simulated Annealing

Recall that we are searching for the minima of a function $E : \Sigma \rightarrow \mathbb{R}$, where $\Sigma := \{-1, +1\}^N$. To find minima of E in dense regions of the state space the authors in [2] propose a new measure given by

$$P_{y,\beta,\gamma}(\sigma) := \frac{\exp(y\Phi_{\beta,\gamma}(\sigma))}{Z(y, \beta, \gamma)} \tag{2}$$

where

$$Z(y, \beta, \gamma) := \sum_{\sigma''} \exp(y\Phi_{\beta,\gamma}(\sigma'')). \tag{3}$$

$P_{y,\beta,\gamma}$ has, at least formally, the structure of a Gibbs measure at inverse temperature y . Its "energy function" is given by

$$\Phi_{\beta,\gamma}(\sigma) := \log \sum_{\sigma' \in \Sigma} \exp(-\beta E(\sigma') - \gamma d(\sigma, \sigma')), \tag{4}$$

where $d(\cdot, \cdot)$ is some monotonically increasing function of a distance on Σ . This distance will be chosen below, but there are not too many reasonable essentially different distance functions on Σ , anyway.

Since $\Phi_{y,\beta,\gamma}$ weights each configuration σ by a function of its distance to σ' and the σ' again by an exponential of their energy, it is, indeed, plausible that $\Phi_{y,\beta,\gamma}$ is much smoother than E and will have its minima in dense regions. We will come back to this question in the next section.

However, a serious problem is, how one could simulate from the measure $P_{y,\beta,\gamma}$. Indeed, computing the "energy" $\Phi_{y,\beta,\gamma}(\sigma)$ of a single configuration σ involves, among others, computing $E(\sigma')$ for all $\sigma' \in \Sigma$. Computing these values is almost as hard as finding the minima of E (even though one might not be immediately able to tell which of these minima are in dense regions). To find a promising algorithm that does not rely on computing all the values of $E(\sigma)$, Baldassi et al. [2] propose the following:

First of all assume that $y \geq 2$ is an integer. Second take as function of the distance between two spins σ and σ' the (negative) inner product: $d(\sigma, \sigma') = -\langle \sigma, \sigma' \rangle$. As a matter of fact, this is a natural choice, since two natural distance functions, the Hamming distance and the square of the Euclidian distance are functions of the inner product: $\sum_i (\sigma_i - \sigma'_i)^2 = 2N - 2\langle \sigma, \sigma' \rangle$ as well as $d_H(\sigma, \sigma') = \frac{N - \langle \sigma, \sigma' \rangle}{2}$ and the N dependent terms cancel, because they also occur

in $Z(y, \beta, \gamma)$. Using the fact that y is an integer, we can now compute the partition function $Z(y, \beta, \gamma)$ (by replacing σ'' by σ in (3)) of this model:

$$\begin{aligned} Z(y, \beta, \gamma) &= \sum_{\sigma \in \Sigma} \exp(y\Phi_{\beta, \gamma}(\sigma)) = \sum_{\sigma \in \Sigma} \exp\left(\sum_{a=1}^y \Phi_{\beta, \gamma}(\sigma^a)\right) \\ &= \sum_{\sigma \in \Sigma} \prod_{a=1}^y \sum_{\sigma^a \in \Sigma} \exp(-\beta E(\sigma^a) - \gamma d(\sigma, \sigma^a)) \\ &= \sum_{\sigma \in \Sigma} \sum_{\sigma^1 \in \Sigma} \dots \sum_{\sigma^y \in \Sigma} \exp(-\beta E(\sigma^1) - \gamma d(\sigma, \sigma^1)) \dots \exp(-\beta E(\sigma^y) - \gamma d(\sigma, \sigma^y)) \\ &= \sum_{\sigma \in \Sigma} \sum_{\{\sigma^a\} \in \Sigma^y} \exp\left(-\beta \sum_{a=1}^y E(\sigma^a) - \gamma \sum_{a=1}^y d(\sigma, \sigma^a)\right). \end{aligned}$$

Here $\sum_{\{\sigma^a\}}$ is the sum over all $\sigma^1, \dots, \sigma^y$. Hence $Z(y, \beta, \gamma)$ can be considered as a partition function on the space of all $(\sigma, \{\sigma^a\}) \in \Sigma^{y+1}$ of the measure

$$Q((\sigma, \{\sigma^a\})) := \frac{\exp(-\beta \sum_{a=1}^y E(\sigma^a) - \gamma \sum_{a=1}^y d(\sigma, \sigma^a))}{Z(y, \beta, \gamma)}. \tag{5}$$

Its marginal with respect to the second coordinate $\{\sigma^a\}$ is given by

$$\bar{Q}(\{\sigma^a\}) := \frac{\sum_{\sigma} \exp(-\beta \sum_{a=1}^y E(\sigma^a) - \gamma \sum_{a=1}^y d(\sigma, \sigma^a))}{Z(y, \beta, \gamma)}. \tag{6}$$

Making use of our choice $d(\sigma, \sigma') = -\langle \sigma, \sigma' \rangle$ we obtain for the numerator in \bar{Q} :

$$\begin{aligned} Z(y, \beta, \gamma) \bar{Q}(\{\sigma^a\}) &= \sum_{\sigma} \exp\left(-\beta \sum_{a=1}^y E(\sigma^a) + \gamma \sum_{a=1}^y \langle \sigma, \sigma^a \rangle\right) \\ &= \frac{2^N}{2^N} \sum_{\sigma} \exp\left(-\beta \sum_{a=1}^y E(\sigma^a) + \gamma \sum_{i=1}^N \sum_{a=1}^y \sigma_i \sigma_i^a\right) \\ &= 2^N \exp\left(-\beta \sum_{a=1}^y E(\sigma^a) + \sum_{i=1}^N \log \cosh\left(\gamma \sum_{a=1}^y \sigma_i^a\right)\right) \end{aligned}$$

Putting the 2^N into the normalizing constant we thus obtain that

$$\bar{Q}(\{\sigma^a\}) = \frac{\exp\left(-\beta \sum_{a=1}^y E(\sigma^a) + \sum_{i=1}^N \log \cosh\left(\gamma \sum_{a=1}^y \sigma_i^a\right)\right)}{Z'(y, \beta, \gamma)}.$$

This form of the measure \bar{Q} is now accessible to a Simulated Annealing algorithm: being in $\{\sigma^a\}$ one picks one of the σ^a at random and one coordinate σ_i^a of σ^a at random and flips it to become $-\sigma_i^a$. This new configuration is then accepted with the usual Simulated Annealing probabilities.

Example 2 (Example 1 continued) In our perceptron example we so far proposed the energy function

$$\bar{E}(W) = - \prod_{\mu=1}^M \Theta(\partial^\mu \langle W, \xi^\mu \rangle) = - \prod_{\mu=1}^{\alpha N} \Theta(\partial^\mu \langle W, \xi^\mu \rangle).$$

This function, however, may be a bit unwieldy when using Simulated Annealing, since it just tells how many patterns have been classified correctly but not whether we are moving in a "good" or a "bad" direction when the proposed configuration $\{W'^a\}$ has the same energy as the old configuration $\{W^a\}$. We therefore propose (as e.g. [2]) to use the energy function

$$E(W) = \sum_{\mu=1}^{\alpha N} E^\mu(W) \quad \text{with } E^\mu(W) = R(-\vartheta^\mu \langle W, \xi^\mu \rangle),$$

instead. Here $R(x) = \frac{x+1}{2} \Theta(x)$ and we again assume that N is odd, otherwise we would need to take $R(x) \frac{x}{2} \Theta(x)$. In other words E^μ is the number of bits that we need to change, in order to classify ξ^μ correctly.

3 Convergence of the Annealing Process

In this section we want to discuss the convergence properties of the annealing procedure introduced above. The two major questions are: Does the process converge to an invariant measure, and if so, does this measure have the desired property of favoring dense regions? This question is not addressed in [2]. However, we feel that it is the first problem that needs to be analyzed. Indeed, if the process does not converge to the desired distribution the question is rather when to stop it than what is the optimal choice of parameters.

We will distinguish two cases: the first is when γ in the definition of the measures Q in (5) and \bar{Q} in (6) does not depend on time, while the second is, when it does.

Before analyzing these two cases, we will slightly modify the annealing procedure, to make it accessible to the best results that are available for discrete time, see [1]. As a matter of fact, we find discrete time slightly more appropriate for computer simulations than the continuous time set-up in e.g. [22], [18], or [8]. To this end, we will study cooling schedules, where the inverse temperature β_n is fixed for T_n consecutive steps of the annealing process. Denote by ν_n the distribution of the annealing process X_n at time

$$L_n := T_1 + \dots + T_n.$$

Note that ν_n can be computed recursively: If S_β denotes the transition matrix of the Metropolis–Hastings chain (see [19] or [21]) at inverse temperature β (see (7) below), then

$$\nu_n = \nu_{n-1} S_{\beta_n}^{T_n} \quad \text{where } \nu_0 \text{ is a fixed probability measure on } \Sigma^y.$$

Here, of course, $S_{\beta_n}^{T_n}$ is the T_n 'th power of the transition matrix S_{β_n} (which is constant for the last T_n steps, as described above).

3.1 Fixed γ

If γ is fixed it is convenient to split the Simulated Annealing algorithm introduced above into a γ -dependent part and a β -dependent part. To this end, let us introduce the following probability measure μ_0 on Σ^y :

$$\mu_0(\{\sigma^a\}) := \frac{\exp\left(\sum_{i=1}^N \log \cosh(\gamma \sum_{a=1}^y \sigma_i^a)\right)}{\Gamma}$$

with $\Gamma := \sum_{\{\tilde{\sigma}^a\}} \exp\left(\sum_{i=1}^N \log \cosh(\gamma \sum_{a=1}^y \tilde{\sigma}_i^a)\right)$.

Next define a transition matrix Π on Σ^y . Π will only allow transition from $\{\sigma^a\}$ to $\{\tilde{\sigma}^a\}$, if there are exactly one σ^a , $a = 1, \dots, y$ and one $i = 1, \dots, N$, such that $\sigma_i^a = -\tilde{\sigma}_i^a$, and for all other b and j we have $\sigma_j^b = \tilde{\sigma}_j^b$. In this case, we define

$$\Pi_\gamma(\{\sigma^a\}, \{\tilde{\sigma}^a\}) := \Pi(\{\sigma^a\}, \{\tilde{\sigma}^a\}) \min\left(1, \frac{\cosh(\gamma \sum_{a=1}^y \tilde{\sigma}_i^a)}{\cosh(\gamma \sum_{a=1}^y \sigma_i^a)}\right).$$

For all other configurations $\{\hat{\sigma}^a\} \neq \{\sigma^a\}$, we have $\Pi_\gamma(\{\sigma^a\}, \{\hat{\sigma}^a\}) = 0$ and we set

$$\Pi_\gamma(\{\sigma^a\}, \{\sigma^a\}) := 1 - \sum_{\{\tilde{\sigma}^a\} \neq \{\sigma^a\}} \Pi_\gamma(\{\sigma^a\}, \{\tilde{\sigma}^a\}).$$

Note that Π_γ is nothing but the Metropolis–Hastings algorithm for the measure μ_0 (see [21]). In particular, Π_γ is reversible with respect to the measure μ_0 , i.e.

$$\mu_0(\{\sigma^a\})\Pi_\gamma(\{\sigma^a\}, \{\tilde{\sigma}^a\}) = \mu_0(\{\tilde{\sigma}^a\})\Pi_\gamma(\{\tilde{\sigma}^a\}, \{\sigma^a\}).$$

Now consider the Metropolis–Hastings chain on Σ^y with proposal chain Π_γ and transition probabilities

$$S_\beta(\{\sigma^a\}, \{\tilde{\sigma}^a\}) := \begin{cases} \exp(-\beta(\sum_{a=1}^y E(\sigma^a) - \sum_{a=1}^y E(\tilde{\sigma}^a))^+) \Pi_\gamma(\{\sigma^a\}, \{\tilde{\sigma}^a\}) & \text{if } \{\sigma^a\} \neq \{\tilde{\sigma}^a\} \\ 1 - \sum_{\{\hat{\sigma}^a\} \neq \{\sigma^a\}} \exp(-\beta(\sum_{a=1}^y E(\sigma^a) - \sum_{a=1}^y E(\hat{\sigma}^a))^+) \Pi_\gamma(\{\sigma^a\}, \{\hat{\sigma}^a\}) & \text{if } \{\sigma^a\} = \{\tilde{\sigma}^a\} \end{cases} \tag{7}$$

Here $(x)^+ := \max\{x, 0\}$. For an appropriate normalizing constant \hat{I} this chain has as its invariant measure

$$\frac{\exp(-\beta \sum_{a=1}^y E(\sigma^a))}{\hat{I}} \mu_0(\{\sigma^a\}) = \overline{Q}(\{\sigma^a\}) =: \overline{Q}_\beta(\{\sigma^a\}). \tag{8}$$

So indeed for each fixed $\beta > 0$, $\gamma > 0$ we have found a Metropolis chain for \overline{Q} .

If we now let $\beta = \beta_n$ depend on n in the form described at the beginning of the section, we arrive at a Simulated Annealing algorithm with piecewise constant temperature.

We will quickly introduce some of Azencott’s notation [1]. The invariant measure of S_{β_n} is \overline{Q}_{β_n} . Recall that we assumed that $\min_{\sigma \in \Sigma} E(\sigma) = 0$ and define

$$B := \min_{\sigma: E(\sigma) \neq 0} E(\sigma). \tag{9}$$

Next we bound

$$\|\overline{Q}_{\beta_n} - \overline{Q}_{\beta_{n+1}}\|_\infty \leq \kappa_1 \exp(-\beta_n B)$$

for some constant κ_1 . Indeed such an estimate is true for any difference of Gibbs measures with respect to the same energy function. To see this let

$$\rho_{\beta_n} := \frac{\exp(-\beta_n U(x))}{Z_n}$$

be a sequence of Gibbs measures with respect to the energy function U on a discrete space of size K . We assume $\min_x U(x) = 0$ (without loss of generality, otherwise we subtract the minimum from U) and $\min_{x: U(x) \neq 0} U(x) = B$. Let $k := |\{x : U(x) = 0\}|$. Then, for any x with $U(x) = 0$ we simply have

$$|\rho_{\beta_n}(x) - \rho_{\beta_{n+1}}(x)| = \frac{|\sum_{y: U(y) \neq 0} \exp(-\beta_n U(y)) - \exp(-\beta_{n+1} U(y))|}{Z_n Z_{n+1}} \leq 2(K - k)e^{-\beta_n B},$$

since $\beta_{n+1} \geq \beta_n$ and $Z_n \geq 1$ for all n . Otherwise, if $U(x) > 0$, we trivially can compute

$$|\rho_{\beta_n}(x) - \rho_{\beta_{n+1}}(x)| \leq e^{-\beta_n U(x)} + e^{-\beta_{n+1} U(x)} \leq 2e^{-\beta_n B}.$$

To describe the spectral gap of S_{β_n} , for any two $\{\sigma^a\}, \{\tau^a\} \in \Sigma^y$ let $\mathcal{P}(\{\sigma^a\}, \{\tau^a\})$ be the set of all paths in Σ^y from $\{\sigma^a\}$ to $\{\tau^a\}$. For $p \in \mathcal{P}(\{\sigma^a\}, \{\tau^a\})$ with vertices $\{v^{a,r}\}_{r=0}^M$ define

$$\text{Elev}(p) := \max_{\{v^{a,r}\}_{r=0}^M} \sum_{a=1}^y E(v^{a,r}).$$

Moreover define

$$H(\{\sigma^a\}, \{\tau^a\}) = \min_{p \in \mathcal{P}(\{\sigma^a\}, \{\tau^a\})} \text{Elev}(p)$$

and

$$m := \max_{\{\sigma^a\}, \{\tau^a\}} (H(\{\sigma^a\}, \{\tau^a\}) - \sum_{a=1}^y E(\sigma^a) - \sum_{a=1}^y E(\tau^a)). \tag{10}$$

The quantity m is related to the optimal cooling schedule for Simulated Annealing as well as to the spectral gap of the associated Metropolis Hastings algorithm S_β . To understand this, define the operator $L_\beta f(\{\sigma^a\})$ for $\{\sigma^a\} \in \Sigma^y$ and $f : \Sigma^y \rightarrow \mathbb{R}$ by

$$L_\beta f(\{\sigma^a\}) = \sum_{\{\tau^a\}} (f(\{\sigma^a\}) - f(\{\tau^a\})) S_\beta(\{\sigma^a\}, \{\tau^a\}).$$

Let \mathcal{E}_β be the associated Dirichlet form, i.e. for functions $f, g : \Sigma^y \rightarrow \mathbb{R}$

$$\begin{aligned} \mathcal{E}_\beta(f, g) &:= - \int (f, L_\beta g) d\bar{Q}_\beta \\ &= \frac{1}{2\hat{\Gamma}} \sum_{\{\sigma^a\}, \{\tau^a\}} (f(\{\sigma^a\}) - f(\{\tau^a\}))(g(\{\sigma^a\}) - g(\{\tau^a\})) \\ &\quad \times \exp\left(-\beta\left(\sum_a E(\sigma^a) \vee \sum_a E(\tau^a)\right)\right) \mu_0(\{\sigma^a\}) \Pi_\gamma(\{\sigma^a\}, \{\tau^a\}). \end{aligned}$$

Then with

$$\psi(\beta) := \inf\{\mathcal{E}(f, f) : \|f\|_{L^2(\bar{Q}_\beta)} = 1 \text{ and } \int f d\bar{Q}_\beta = 0\}$$

we have

Proposition 1 *There are constants $c > 0$ and $C < \infty$ such that for all $\beta \geq 0$,*

$$c e^{-\beta m} \leq \psi(\beta) \leq C e^{-\beta m}.$$

Proof See [22, Theorem 2.1]. □

But we also have that

$$\psi(\beta) = 1 - \lambda_1(\beta)$$

where $\lambda_1(\beta)$ is the second largest eigenvalue of S_β , cf. [23, p. 176] or [9, (1.2)]. This establishes the relation of m to the spectral gap of S_β .

Introduce

$$\varepsilon_n := \|\bar{Q}_{\beta_n} - \nu_n\|_\infty.$$

Then

Theorem 1 ε_n converges to 0, if

$$\lim_{n \rightarrow \infty} \left[-\sum_{k=1}^n T_k \exp(-\beta_k m) + n \log \kappa_1 \right] = -\infty. \tag{11}$$

In particular, we need that $\sum_{k=1}^n T_k \exp(-\beta_k m) \rightarrow \infty$. In this case ν_n has the same limit as \bar{Q}_{β_n} and this is given by a distribution \bar{Q}_∞ on

$$\mathcal{N}_0 := \{\{\sigma^a\} : \sum_{a=1}^y E(\sigma^a) = 0\} \tag{12}$$

such that

$$\bar{Q}_\infty(\{\sigma^a\}) \asymp \mu_0(\{\sigma^a\}), \quad \text{if } \{\sigma^a\} \in \mathcal{N}_0$$

and $\bar{Q}_\infty(\{\sigma^a\}) = 0$, otherwise (here \asymp denotes proportionality). Of course \bar{Q}_∞ is normalized in such a way that it is a probability measure on \mathcal{N}_0 .

Proof The convergence part is basically the content of [1, Sect. 7]. Note that the computations there are done for a proposal chain Π that has the uniform measure as its invariant distribution. However, the proof on p. 231 [1] carries over verbatim to our situation.

After that it is easy matter to check that \bar{Q}_{β_n} has a limiting distribution \bar{Q}_∞ and that \bar{Q}_∞ charges every point in \mathcal{N}_0 with a probability proportional to $\mu_0(\{\sigma^a\})$. \square

A choice for T_k where (11) holds is given by

$$T_k := \frac{\exp(-m\beta_k)}{C} (\log \kappa_1 + Cb)$$

for a constant $b > 0$, m as given in (10), and C as given in Proposition 1.

As Azencott [1] points out, in this case

$$\beta_n \sim \frac{\alpha}{B} \log n + \frac{b}{B} n,$$

for some $\alpha > 1$ and B defined as in (9),

$$T_n \sim n^{\frac{\alpha m}{B}} \exp\left(\frac{bB}{m} n\right),$$

and T_n and L_n have the same order of magnitude, i.e. the algorithm spends most of the time in the lowest temperature band.

One also sees the logarithmic relation between β and T , i.e.

$$\beta_n \sim \frac{1}{m} \log(L_n) \sim \frac{1}{m} \log(T_n).$$

We now turn to the question whether this algorithm achieves that typical samples from it have realizations in dense regions of Σ . First of all this needs to be defined:

Definition 1 Let $\sigma \in \Sigma$ with $E(\sigma) = 0$ and let $R > 0$ and $k \in \mathbb{N}$. The discrete ball $B_R(\sigma) \subset \Sigma$ with radius R , centered in σ is called an (R, k) -dense with respect to E , if there are exactly k global minima τ of E in $B_R(\sigma)$. (Without loss of generality all balls considered here and henceforth are Hamming balls.)

σ is called R -isolated, if σ is the only global minimum of E in $B_R(\sigma)$.

The authors in [2] are not very explicit about a definition of "dense regions" and the situation where Replicated Simulated Annealing should be applied. However, from their examples, they seem to have in mind a situation close to the following caricature:

Situation 1 Given $1 < a < b < 1$ and $\alpha_N \rightarrow \infty, \delta_N \rightarrow \infty$ with

$$\lim_{N \rightarrow \infty} \frac{\alpha_N}{\delta_N} = 0 \quad \text{as well as} \quad \lim_{N \rightarrow \infty} \frac{\delta_N}{N} = 0,$$

we say that a sequence of energy functions E_N on $\Sigma_N := \Sigma = \{-1, +1\}^N$ is $(a, b, \alpha_N, \delta_N)$ -regular, if it has b^N global minima, if there exists $\sigma \in \Sigma_N$ such that $B_{\alpha_N}(\sigma)$ is (α_N, a^N) -dense and such that all the other $b^N - a^N$ minima are δ_N -isolated.

It is now rather obvious that $\bar{Q}_\infty(\{\cdot\})$ prefers such dense regions:

Proposition 2 Assume we are in the situation described in Situation 1. Hence we have a sequence of energy functions that is $(a, b, \alpha_N, \delta_N)$ -regular. Then, given $\varepsilon > 0$, for any admissible choice of these parameters, there exist y, N_0 and γ such that

$$\bar{Q}_\infty(\times_{i=1}^y B_{\alpha_N}(\sigma)) := \bar{Q}_\infty(\{(\sigma^1, \dots, \sigma^y) : \sigma^a \in B_{\alpha_N}(\sigma) \forall a\}) \geq 1 - \varepsilon$$

for all $N \geq N_0$.

Proof Note that \bar{Q}_∞ has its mass concentrated on the set \mathcal{N}_0 (given by equation (12)) and the differences in the mass for the various configurations from this set stem from factor

$$\mu_0(\{\sigma^a\}) = \frac{\exp\left(\sum_{i=1}^N \log \cosh(\gamma \sum_{a=1}^y \sigma_i^a)\right)}{\Gamma}$$

Let us just consider the numerators of these weights.

Let σ be an δ_N -isolated minimum of E_N . If all $\sigma^1, \dots, \sigma^y$ are located in σ , then the numerator of $\mu_0(\{\sigma^a\})$ equals $\exp(N \log \cosh(\gamma y))$. Otherwise there is at least one σ^a that is different from σ , say in a global minimum τ of E_N . By assumption $d_H(\sigma, \tau) \geq \delta_N$. Thus a configuration that has at least one $\sigma^a = \tau$ has a weight at most

$$\exp((N - \delta_N) \log \cosh(\gamma y) + \delta_N \log \cosh((y - 2)\gamma)).$$

Now there are $b^N - a^N$ δ_N isolated minima. Hence the sum of the numerators of the probabilities of these isolated minima can be bounded from above by

$$\exp(N \log \cosh(\gamma y)) b^N \left(1 + b^{Ny} \frac{e^{(N - \delta_N) \log \cosh(\gamma y) + \delta_N \log \cosh((y - 2)\gamma)}}{e^{N \log \cosh(\gamma y)}} \right).$$

Here b^N is a bound on the number of isolated minima, $\exp(N \log \cosh(\gamma y))$ is the weight, when all σ^a are identical, b^{Ny} is an upper bound on the number of choices we have, when one σ^a equals a given isolated minimum and at least one σ^b is different, and finally $e^{(N - \delta_N) \log \cosh(\gamma y) + \delta_N \log \cosh((y - 2)\gamma)}$ is a rough upper bound on the weight in that case.

Note, that we will choose γ and y below in such a way that $y\gamma \rightarrow \infty$, when $N \rightarrow \infty$. We will therefore bound $\log \cosh(y\gamma) \leq y\gamma$. Then the total contribution of the isolated minima becomes at most:

$$e^{N\gamma y} b^N \left(1 + b^{Ny} e^{-2\gamma\delta_N} \right)$$

If we choose $\gamma \geq \frac{Ny \log b}{2\delta_N}$ the contribution to the numerator of the probability of the isolated minima will be at most $2 \exp(N\gamma y) b^N$.

On the other hand, for the case that all σ^a are in the dense region $B_{\alpha_N}(\sigma)$ we have a^{Ny} choices. For each of these choices at least $N - y\alpha_N$ of the coordinates of all $\sigma^1, \dots, \sigma^y$ are identical. Again, since $y\gamma \rightarrow \infty$, when $N \rightarrow \infty$, given $\varepsilon' > 0$ we may bound $\log \cosh(y\gamma) \geq y\gamma(1 - \varepsilon')$. Thus the overall weight (this is again the numerator of the corresponding probability) of the dense region is at least $a^{Ny} e^{(N-y\alpha_N)\gamma y(1-\varepsilon')}$.

To compare the two weights, let us see, if we can arrange the parameters in such a way that

$$a^{Ny} e^{(N-y\alpha_N)\gamma y(1-\varepsilon')} \gg 2e^{N\gamma y} b^N$$

(by which we mean that

$$\frac{a^{Ny} e^{(N-y\alpha_N)\gamma y(1-\varepsilon')}}{2e^{N\gamma y} b^N} \rightarrow \infty$$

as $N \rightarrow \infty$). Since $\varepsilon' > 0$ is fixed and arbitrarily small, we may as well check whether

$$a^{Ny} e^{(N-y\alpha_N)\gamma y} \gg 2e^{N\gamma y} b^N$$

which is the case, if and only if

$$\exp \left(Ny \left(\log a - \frac{y\gamma\alpha_N}{N} - \frac{\log b}{y} \right) \right) \gg 1.$$

To this end, substitute $\gamma = \frac{Ny \log b}{2\delta_N}$ (and note that indeed $\gamma = \gamma_N \rightarrow \infty$ as $N \rightarrow \infty$) to obtain for the exponent on the right hand side:

$$Ny \left(\log a - \frac{y^2 \log b \alpha_N}{2\delta_N} - \frac{\log b}{y} \right).$$

Now take $y = \lceil \frac{1}{2} \frac{\log b}{\log a} \rceil$. Since, by assumption $\frac{\alpha_N}{\delta_N} \rightarrow 0$ and y does not depend on N , also $\frac{y^2 \log b \alpha_N}{2\delta_N}$ converges to 0. This implies that the exponent will eventually become negative, hence the dense region carries an arbitrarily large mass. \square

Remark 1 Reading [2] carefully, one may get the impression that for them a dense region is one with an exponential number of local minima of E_N (again, the authors in [2] are not very explicit about this). However, if we are taking the limit $\beta \rightarrow \infty$ slowly enough as in a real Simulated Annealing schedule, the local minima that are not global minima will eventually get zero probability and hence are negligible. As a matter of fact, if one works with finite times as in our next section, this is, of course, not true. In this case however, one could equally well study a low temperature Metropolis chain, since most of the time in the annealing schedules is spent in the low temperature region, anyway, as remarked above. For this Metropolis–Hastings chain a result similar to Proposition 2 can be shown very similarly.

3.2 The Limit $\gamma \rightarrow \infty$

The situation where also γ depends on time and converges to infinity, when time becomes large, is different to the fixed γ situation. Even though this is not explicitly stated in [2] it seems to be the version of the algorithm that the authors in have in mind. Indeed, as mentioned, they only consider a finite time horizon, in which they, however, increase γ .

In the situation with $\gamma \rightarrow \infty$ we need to modify the considerations of the previous section. Again we will assume that we keep β_n, γ_n constant on an interval $T_n \leq t \leq T_{n+1} - 1$. For the algorithm in this fixed time interval, again, the invariant measure is given by \bar{Q}_β with $\beta = \beta_n$ and $\gamma = \gamma_n$ as given in (8). This is the case because during this interval the parameters of the Metropolis chain do not change. To stress the dependence on both parameters, we will now denote this measure by $\bar{Q}_{\beta,\gamma}$.

Following the arguments in the previous subsection we now see that there is a constant κ_2 , such that

$$\|\bar{Q}_{\beta_n,\gamma_n} - \bar{Q}_{\beta_{n+1},\gamma_{n+1}}\|_\infty \leq \kappa_2 e^{-(\beta_n B + \gamma_n B')}.$$

Here again, $B := \min_{\sigma: E(\sigma) \neq 0} E(\sigma)$. Analogously, the constant B' is defined as the gap between the maximum of the function

$$H(\{\sigma^a\}) := \sum_{i=1}^N \log \cosh\left(\sum_{a=1}^y \sigma_i^a\right) \quad \text{on } \Sigma^y$$

and its second largest value. Hence

$$\begin{aligned} B' &:= N \log \cosh(\gamma y) - ((N - 1) \log \cosh(\gamma y) + \log \cosh(\gamma(y - 2))) \\ &= \log \cosh(\gamma y) - \log \cosh(\gamma(y - 2)). \end{aligned}$$

The maximum of H is realized when we take all σ^a identical, while the second term in B' stems from the fact that the we obtain the second largest value of H by changing one σ^a in one spin from a maximizing configuration. Since we will consider the limit $\gamma \rightarrow \infty$ we may safely replace $\log \cosh(\gamma y)$ by $\gamma y - \log 2$ and $\log \cosh(\gamma(y - 2))$ by $\gamma(y - 2) - \log 2$ to obtain

$$B' \approx 2\gamma.$$

To determine how the cooling schedule has to be chosen, we need to estimate the spectral gap of the Metropolis chain. Note that, if we use Proposition 1 to do so, we run into the problem, that the constants c and C there depend on time, because the energy function does. The solution is, of course, to include this time dependence into the definitions. Hence for a time t let

$$F_t(\{\sigma^a\}) := \sum_{a=1}^y E(\sigma^a) - \frac{\gamma_t}{\beta_t} H(\{\sigma^a\}).$$

Then, we can represent the Simulated Annealing chain, which we will now denote by $S_{\beta,\gamma}$ and which is still given by (7) (with the only difference that now also Π_γ depends on time) as a Simulated Annealing algorithm with time-dependent energy function F_t , see e.g. [28], [14]. Indeed, in this case we may replace the proposal chain Π_γ in (7) to Π . Here Π being in $\{\sigma^a\}$ picks one of $a = 1, \dots, y$ and one index $i = 1, \dots, N$ at random and flips σ_i^a to $-\sigma_i^a$. Then $S_{\beta,\gamma}$ can be written as

$$\begin{aligned}
 & S_{\beta,\gamma}(\{\sigma^a\}, \{\tilde{\sigma}^a\}) \\
 & := \begin{cases} \exp(-\beta(F_t(\{\sigma^a\}) - F_t(\{\tilde{\sigma}^a\}))^+) \Pi(\{\sigma^a\}, \{\tilde{\sigma}^a\}) & \text{if } \{\sigma^a\} \neq \{\tilde{\sigma}^a\} \\ 1 - \sum_{\{\hat{\sigma}^a\}} \exp(-\beta(F_t(\{\sigma^a\}) - F_t(\{\hat{\sigma}^a\}))^+) \Pi(\{\sigma^a\}, \{\hat{\sigma}^a\}) & \text{if } \{\sigma^a\} = \{\tilde{\sigma}^a\} \end{cases}
 \end{aligned} \tag{13}$$

Now we can use results from [28] (cf. [27] for related work) to compute the spectral gap $S_{\beta,\gamma}$. In analogy to what we did in the previous subsection, define

$$m_t := \max_{\{\sigma^a\}, \{\tau^a\}} (H_t(\{\sigma^a\}, \{\tau^a\}) - F_t(\{\sigma^a\}) - F_t(\{\tau^a\}))$$

where

$$H_t(\{\sigma^a\}, \{\tau^a\}) = \min_{p \in \mathcal{P}(\{\sigma^a\}, \{\tau^a\})} \text{Elev}_t(p)$$

and

$$\text{Elev}_t(p) := \max_{\{v^a, r\}_{r=0}^M} F_t(\{v^a\}).$$

Again, in analogy to the previous subsection for functions $f, g : \Sigma^y \rightarrow \mathbb{R}$ let the Dirichlet-form $\mathcal{E}_{\beta,\gamma}$ be given by

$$\begin{aligned}
 \mathcal{E}_{t,\beta,\gamma}(f, g) &= \frac{1}{2\hat{r}} \sum_{\{\sigma^a\}, \{\tau^a\}} (f(\{\sigma^a\}) - f(\{\tau^a\}))(g(\{\sigma^a\}) - g(\{\tau^a\})) \\
 &\quad \times \exp(-\beta(F_t(\{\sigma^a\}) \vee F_t(\{\tau^a\}))) \mu_0(\{\sigma^a\}) \Pi(\{\sigma^a\}, \{\tau^a\}).
 \end{aligned}$$

Then for

$$\psi_t(\beta) := \inf\{\mathcal{E}_{t,\beta,\gamma}(f, f) : \|f\|_{L^2(\bar{Q}_{\beta,\gamma})} = 1 \text{ and } \int f d\bar{Q}_{\beta,\gamma} = 0\}$$

it holds

Proposition 3 *There are constants $c > 0$ and $C < \infty$ such that for all $\beta \geq 0$.*

$$ce^{-\beta m_t} \leq \psi_t(\beta) \leq Ce^{-\beta m_t}.$$

Proof This is the content of [28, Theorem 2.1]. □

As before Proposition 3 implies for the second largest eigenvalue $\lambda_1(\beta, \gamma)$ of $S_{\beta,\gamma}$ that

$$|\lambda_1(\beta_n, \gamma_n)| \leq 1 - Ce^{-\beta_n m_N}.$$

From linear algebra we therefore obtain that for each $n = 1, 2, \dots$ and any probability measure ν_0 on Σ^y

$$\|\nu_0 S_{\beta_n, \gamma_n}^{T_n}\|_\infty \leq \kappa_3 (1 - Ce^{-\beta_n m_n})^{T_n} \|\nu_0\|_\infty$$

for some constant $\kappa_3 > 0$ (cf. the very similar argument for ordinary Simulated Annealing in [1, (7.8)]). Writing again

$$\varepsilon_n := \|\bar{Q}_{\beta_n, \gamma_n} - \nu_n\|_\infty$$

by the recursive structure of the annealing algorithm and the considerations above we obtain the estimate

$$\varepsilon_n \leq \kappa_2 e^{-(\beta_n B + \gamma_n B')} + \kappa_3 (1 - Ce^{-\beta_n m_n})^{T_n} \varepsilon_{n-1}.$$

Solving this recursive inequality gives

$$\varepsilon_n \leq \kappa_3^n \prod_{k=1}^n (1 - Ce^{-\beta_k m_k})^{T_k} \left(\sum_{k=1}^n \frac{\kappa_2 e^{-(\beta_n B + \gamma_n B')}}{u_k} + \varepsilon_0 \right) \tag{14}$$

(cf. [1, (7.14)]). Here

$$u_k := \kappa_3^k \prod_{j=1}^k (1 - Ce^{-\beta_j m_j})^{T_j}.$$

Hence we need to chose our parameters β_n, γ_n, T_n in such a way that the right hand side converges to zero. In this case we have shown the following theorem.

Theorem 2 *If*

$$\kappa_3^n \prod_{k=1}^n (1 - Ce^{-\beta_k m_k})^{T_k} \left(\sum_{k=1}^n \frac{\kappa_2 e^{-(\beta_n B + \gamma_n B')}}{u_k} + \varepsilon_0 \right) \rightarrow 0 \tag{15}$$

as $n \rightarrow \infty$, the distribution ν_n of S_{β_n, γ_n} has the same limit as $\bar{Q}_{\beta_n, \gamma_n}$ as $n \rightarrow \infty$.

Following [1, (7.17)] a necessary condition for (15) is

$$\lim_{n \rightarrow \infty} \left(- \sum_{k=1}^n T_k e^{-\beta_k m_k} + n \log \kappa_3 \right) = -\infty$$

Remark 2 If we are right with the assumption that the authors in [2] would take $\gamma_n \rightarrow \infty$ when time gets large, the result of the theorem is, however, not what the authors in [2] seem to intend with their introduction of Replicated Simulated Annealing algorithm. Indeed, when $\beta_n \rightarrow \infty$, and $\gamma_n \rightarrow \infty$ the measure $\bar{Q}_{\beta_n, \gamma_n}$ converges to $\bar{Q}_{\infty, \infty}$. However, the latter is nothing but the uniform distribution on

$$\tilde{\mathcal{N}}_0 := \{ \{\sigma^1, \dots, \sigma^y\} : E(\sigma^a) = 0 \text{ for all } a = 1, \dots, y, \text{ and } \sigma^1 = \dots = \sigma^y \}.$$

In particular, $\bar{Q}_{\infty, \infty}$ does not put higher probability on configuration in dense regions of the state space.

Remark 3 Note that for both, Theorems 1 and 2, the cooling schedules have to be chosen very carefully. An anonymous referee remarked that there are simulation algorithms for Gibbs measures that do not use such a cooling strategy as parallel tempering [33], swapping [16,17,32], or equi-energy sampling [26]. We are grateful for this remark.

However, there are some issues with these algorithms. First of all, all of these algorithms simulate Gibbs measures at non-zero temperatures. That means we will obtain an impression of the energy landscapes, but not necessarily convergence towards their global maxima or minima. However, for the simulations in Sect. 4 this is still an important remark.

The tempering algorithms usually suffer from the deficit that they require computation of partition functions which is as hard as finding the minima or maxima of the energies involved. Swapping circumvents these problems. However, the speed convergence may be a problem (as it is for simulated annealing). In some situations the swapping algorithm converges rapidly (i.e. in polynomial time), see e.g. [12,29,31], in others the convergence takes exponentially long, see [3] or [10]. The results in [11] show that equi-energy sampling typically does not overcome the problem of torpid mixing.

In the next section, we empirically study a slightly modified version of the algorithm of Replicated Simulated Annealing, using both synthetic toy datasets and real data bases.

4 Experiments

Throughout this section, we present various experiments we conducted to empirically study the effectiveness of Replicated Simulated Annealing. While the previous section had an emphasis on theoretical results on the asymptotic behaviour of the algorithm—which from our point of view is necessary for its introduction—the current section analyzes its finite time behaviour and the role of the choice of parameters.

Notice that hence in this simulation section we will necessarily stay closer to the setting in [2]. Especially, other than in the preceding theoretical section we will not let β and γ tend to infinity (for the theoretical part this was necessary in order to get convergence results, while it is impossible in practical applications). The precise setting will be described below. We will put an emphasis on studying the effect of the choice of these hyperparameters on the performance of our algorithm, as well as the robustness of the found solutions.

We conduct our experiments using the MNIST dataset, also described below, and synthetic data.

4.1 MNIST Dataset

MNIST is a dataset of images depicting digits between 0 and 9. We randomly choose a learning set of 6000 examples per digit, i.e. these examples are used to calibrate the model. The aim is to train a classifier to correctly predict which digits are depicted in previously unseen images. This ability of generalization is measured using a test set containing 1000 examples per digit, distinct from those appearing in the training set. The proportion of correctly classified images in the training set (resp. test set) is called the training accuracy (resp. test accuracy). MNIST images are 28×28 pixels and grey-leveled. As such, they are typically represented by a 784-sized vector of numbers between 0 and 255.

When training a binary (weights can only be -1 or 1) logistic regression classifier on MNIST using Replicated Simulated Annealing, we typically achieve a 88% accuracy on the test set, which is on par with the performance obtained with continuous weights and gradient descent. Note that when training our models, we use the cross-entropy loss as our energy, which we refer to as the training loss in the following. In the case of classification with $K = 10$ classes, the output of the model associated to an input x_i is a probability vector $y_i = (y_{i,1}, \dots, y_{i,K})$, and the cross-entropy is then

$$H(t, y) = - \sum_{i=1}^n \sum_{k=1}^K t_{i,k} \log(y_{i,k}),$$

where $t_i \in \{0, 1\}^K$ is the true class of x_i . Beside, the cross-entropy loss on the test set is referred to as the test loss. We train the networks for a total of 300,000 total iterations, starting from a random configuration. Our models contain a total of $784 \cdot 10 = 7840$ parameters, corresponding to a single matrix the input of which is a raw image of 784 dimensions and the output of which is a 10-sized vector where the largest coordinate indicates the associated decision.

4.2 Effect of the Initial and Final Values of β

As mentioned above in our experiments we will always take β from a certain bounded range of values $[\beta_i, \beta_f]$ (these bounds will be used in the remaining of this work). We first explore

Table 1 Number of active transitions, during the learning of MNIST, as a function of β_i and β_f

β_i/β_f	10	100	1000	10,000	100,000	1,000,000
1	271,733	246,793	204,681	160,424	129,138	107,549
10	251,891	219,362	172,511	122,851	92,480	74,444
100	220,294	182,280	121,842	73,605	48,810	36,684
1000	171,610	122,934	60,901	24,248	13,690	10,216
10,000	125,211	76,515	25,593	7542	5061	4415

Table 2 Final MNIST train set loss and corresponding accuracies, as a function of β_i and β_f

β_i/β_f	100	1000	10,000	100,000	1,000,000
1	3.49 (77.56%)	1.71 (86.08%)	1.47 (87.57%)	1.43 (87.53%)	1.47 (87.35%)
10	3.54 (76.84%)	1.72 (86.10%)	1.44 (87.29%)	1.44 (87.44%)	1.44 (87.41%)
100	3.32 (77.66%)	1.64 (86.34%)	1.35 (87.78%)	1.32 (87.56%)	1.37 (87.64%)
1000	3.18 (78.60%)	1.59 (86.00%)	1.45 (87.25%)	1.52 (87.03%)	1.50 (86.94%)
10,000	3.14 (77.57%)	1.69 (85.07%)	1.61 (86.16%)	1.59 (86.95%)	1.59 (86.70%)

Bold numbers correspond to the best results across the table

the influence of the initial and final values for β . Throughout our experiments, we change the value of β from β_i to β_f following an exponential interpolation where $\beta = \beta_i (\beta_f / \beta_i)^{it/it_{\max}}$, it being the current number of iterations and it_{\max} the total number of iterations. This choice of interpolation appeared to give the best and most consistent results among the interpolations we tried, including linear and quadratic with various parameters. Note that for this first series of experiments we only train one model ($\gamma = 0$). First in Table 1, we indicate the number of active transitions (when a potential flip of a value has been performed). Little surprisingly, we observe that the higher the values of β , the less likely we perform flips. We observe a range of two orders of magnitude with our selected parameters.

In Table 2, we depict the corresponding training loss and training accuracy. Interestingly, we observe that the largest values of β are not necessarily giving the best results, suggesting that allowing to perform flips that immediately slightly lower the loss can be beneficial in the long run. We also observe that the results do not seem to be very sensitive of the choice of the initial and final values for β , as a large range of these values yield a very similar performance. Together with Table 1, we can observe that $\beta_i = 100$ and $\beta_f = 100,000$ is a reasonable choice of parameters. This is also confirmed by the results given in Table 3 where we depict the corresponding test loss and test accuracy.

4.3 Influence of γ

To gain a better understanding of the influence of γ , in the next series of experiments we reduce the number of training samples to accelerate computations. Namely we use 10,000 arbitrary training samples. We perform 10 runs for each value of γ , choosing the best values of β_i and β_f found in the previous section. We plot the error bars (confidence interval at 95%) for each value of γ . In Fig. 1 we depict the evolution of the training accuracy and training loss. In Fig. 2 the evolution of the test accuracy and test loss, and in Fig. 3 the evolution of the number of active transitions. We observe that γ helps in finding better solutions, that is

Table 3 Final MNIST test set loss and corresponding accuracies, as a function of β_i and β_f

β_i/β_f	100	1000	10,000	100,000	1,000,000
1	3.27 (78.29%)	1.69 (86.66%)	1.51 (87.50%)	1.55 (87.46%)	1.56 (87.51%)
10	3.41 (77.45%)	1.72 (85.94%)	1.50 (87.62%)	1.54 (87.34%)	1.52 (87.35%)
100	3.05 (78.84%)	1.60 (86.31%)	1.51 (87.48%)	1.36 (87.69%)	1.46 (87.46%)
1000	3.11 (78.88%)	1.68 (86.31%)	1.61 (87.11%)	1.63 (87.09%)	1.65 (87.14%)
10,000	3.13 (77.81%)	1.69 (85.77%)	1.65 (86.69%)	1.73 (86.55%)	1.68 (86.46%)

Bold numbers correspond to the best results across the table

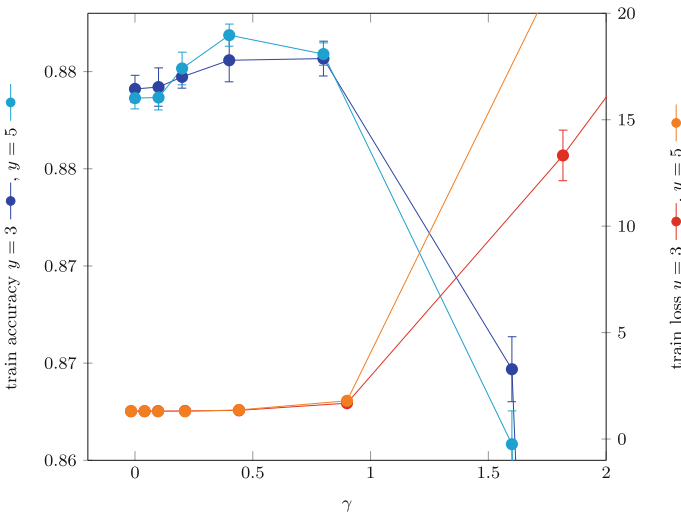


Fig. 1 Evolution of the train accuracy and train loss as a function of γ , and for various values of y

to say solutions with higher accuracies on both the training and the test set. That is only true for a limited range though, as increasing γ too much lead to dramatic decrease in overall performance. This is not surprising as a too large γ leads to forbid many transitions that would result in reducing the loss. Also this may be seen as being in agreement with the findings of Proposition 2, Theorem 15 and Remark 2 in the previous section (even though there we chose the parameters in such a way to convergence to an invariant measure was guaranteed).

4.4 Robustness of Trained Models

To study the robustness of trained models, we consider randomly perturbing a proportion p of the weights in the trained models, and evaluating the impact on the test accuracy. We average each point over 1000 runs of random perturbations, but since it takes a very long time to train the models with MNIST, we always use the same trained models (one for each value of γ). In Fig. 4, we depict the results for $y = 3$, in Fig. 5 for $y = 5$, and in Fig. 6 for $y = 7$. In order to add statistically more significant results, we also plot in Fig. 7 results obtained with synthetic data and $y = 10$. Synthetic data is created by generating 30 vectors uniformly drawn with repetition from all binary vectors of size 100. In this experiment, we

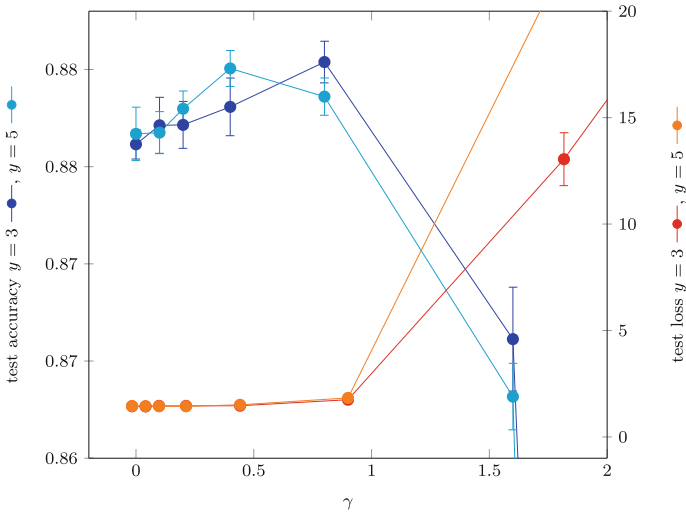


Fig. 2 Evolution of the test accuracy and test loss as a function of γ , and for various values of y

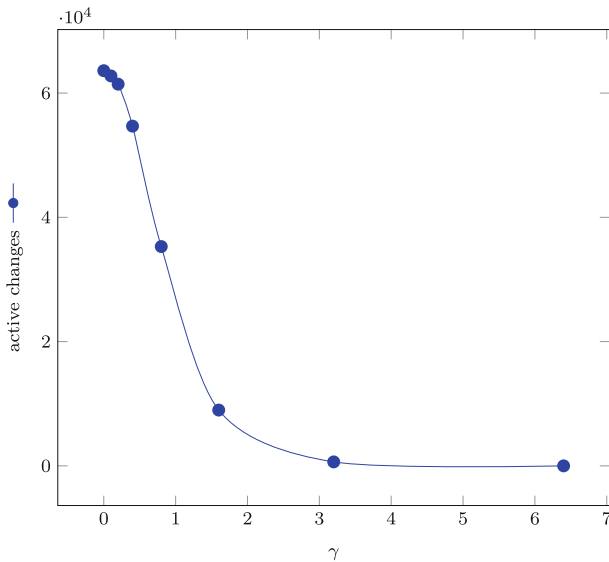


Fig. 3 Evolution of the number of active iterations as a function of γ , and for various values of y

average the results over 1000 tests for each point. For this additional experiment, we found that the best values are $\beta_i = 0.1$ and $\beta_f = 1000$.

Interestingly, we observe that the most robust models are the ones for a balanced value of γ , typically 0.8 or 1.6. This is even true for the case of synthetic data, despite the fact all the models start with a perfect accuracy of 100% when uncorrupted. This is inline with the claims of the authors of [2].

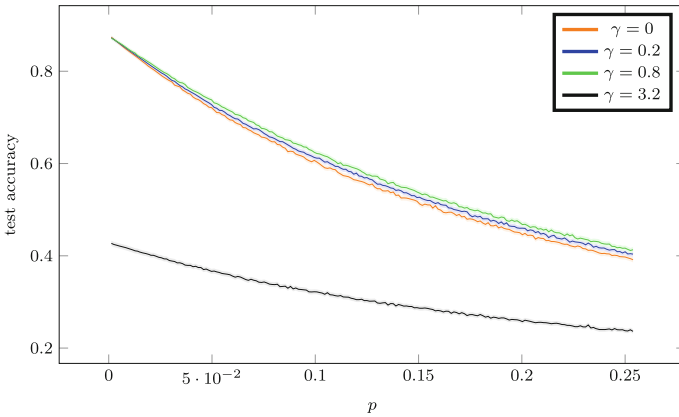


Fig. 4 Robustness of trained models on MNIST as a function of the proportion of flipped parameters p ($y = 3$). Shaded regions around the curves correspond to the confidence interval at 95%

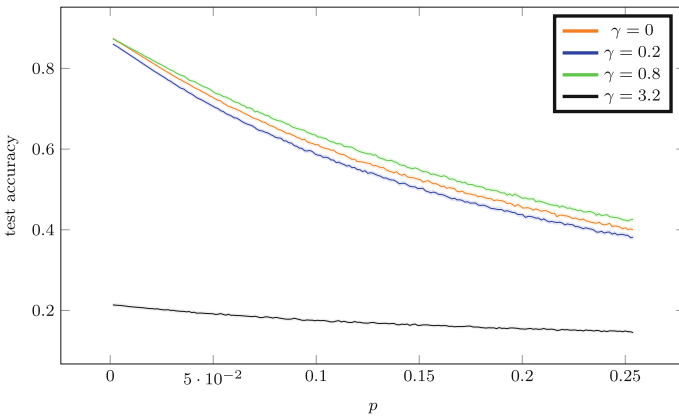


Fig. 5 Robustness of trained models on MNIST as a function of the proportion of flipped parameters p ($y = 5$). Shaded regions around the curves correspond to the confidence interval at 95%

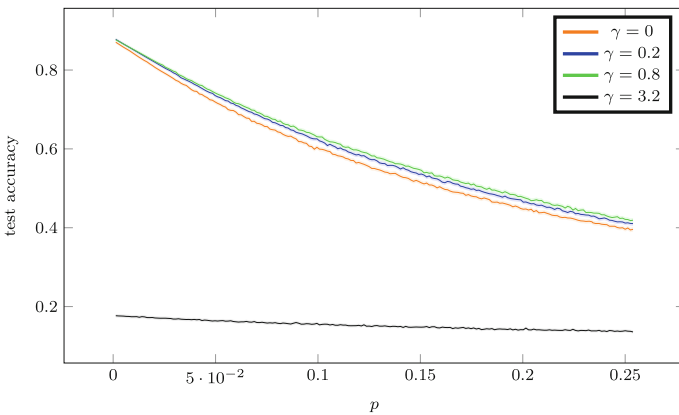


Fig. 6 Robustness of trained models on MNIST as a function of the proportion of flipped parameters p ($y = 7$). Shaded regions around the curves correspond to the confidence interval at 95%

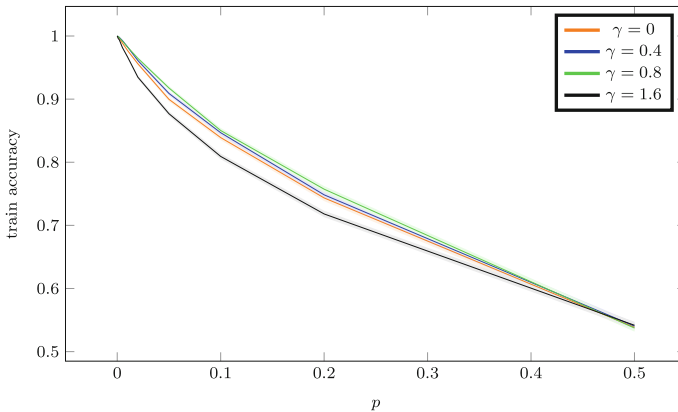


Fig. 7 Robustness of trained models with synthetic data as a function of the proportion of flipped parameters p ($y = 10$). Shaded regions around the curves correspond to the confidence interval at 95%

5 Conclusion

In this work, we have proposed to mathematically and empirically study the algorithm of Replicated Simulated Annealing, that is used to find good configurations of discrete weights neural networks. Here the term “good configurations” refers to configurations in so called dense regions. We have proposed a definition of such dense regions, which are supposed to yield good generalization properties. We have given conditions that ensure convergence of the algorithm and discussed its ability to find good configurations in dense robust regions of the search space. We have seen that to do so the parameter β always need to be taken to infinity when time becomes large, while the parameter γ needs to stay finite.

We also performed experiments using both real datasets and synthetic data to illustrate the role of the choice of the parameters in finite time. Overall, our findings show that Replicated Simulated Annealing is able to find interesting, i.e. “good”, configurations, but that the gain compared to a simple Simulated Annealing is rather small, sometimes even nonexistent in the asymptotic regime, depending on whether one lets $\gamma \rightarrow \infty$ or not.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Azencott, R.: Simulated annealing. *Astérisque*, (161–162): Exp. No. 697, 5, 223–237 (1989), 1988. Séminaire Bourbaki, vol. 1987/88

2. Baldassi, C., Borgs, C., Chayes, J.T., Ingrosso, A., Lucibello, C., Saglietti, L., Zecchina, R.: Unreasonable effectiveness of learning neural networks: from accessible states and robust ensembles to basic algorithmic schemes. *Proc. Natl. Acad. Sci.* **113**(48), E7655–E7662 (2016)
3. Bhatnagar, N., Randall, D.: Torpid mixing of simulated tempering on the Potts model. In: *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 478–487 (electronic), New York. ACM (2004)
4. Biroli, G., Cammarota, C., Ricci-Tersenghi, F.: How to iron out rough landscapes and get optimal performances: averaged gradient descent and its application to tensor PCA. *J. Phys. A* **53**(17), 174003 (2020)
5. Catoni, O.: Rough large deviation estimates for simulated annealing: application to exponential schedules. *Ann. Probab.* **20**(3), 1109–1146 (1992)
6. Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I.-J., Srinivasan, V., Gopalakrishnan, K.: Pact: parameterized clipping activation for quantized neural networks. arXiv preprint [arXiv:1805.06085](https://arxiv.org/abs/1805.06085) (2018)
7. Courbariaux, M., Bengio, Y., David, J.-P.: Binaryconnect: training deep neural networks with binary weights during propagations. In: *Advances in Neural Information Processing Systems*, pp. 3123–3131 (2015)
8. Deuschel, J.-D., Mazza, C.: L^2 convergence of time nonhomogeneous Markov processes. I. Spectral estimates. *Ann. Appl. Probab.* **4**(4), 1012–1056 (1994)
9. Diaconis, P., Stroock, D.: Geometric bounds for eigenvalues of Markov chains. *Ann. Appl. Probab.* **1**(1), 36–61 (1991)
10. Ebbers, M., Löwe, M.: Torpid mixing of the swapping chain on some simple spin glass models. *Markov Process. Relat. Fields* **15**(1), 59–80 (2009)
11. Ebbers, M., Löwe, M.: Equi-energy sampling does not converge rapidly on the mean-field potts model with three colors close to the critical temperature. *J. Phys. A* **53**(11), 115001 (2020)
12. Ebbers, M., Knöpfel, H., Löwe, M., Vermet, F.: Mixing times for the swapping algorithm on the Blume-Emery-Griffiths model. *Random Struct. Algorithms* **45**(1), 38–77 (2014)
13. Esser, S.K., McKinstry, J.L., Bablani, D., Appuswamy, R., Modha, D.S.: Learned step size quantization. arXiv preprint [arXiv:1902.08153](https://arxiv.org/abs/1902.08153) (2019)
14. Frigerio, A., Grillo, G.: Simulated annealing with time-dependent energy function. *Math. Z.* **213**(1), 97–116 (1993)
15. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**(6), 721–741 (1984)
16. Geyer, C.J.: Markov chain monte carlo maximum likelihood. In: *Computing Science and Statistics: Proceedings of 23rd Symposium on the Interface Interface Foundation*, pp. 156–163. Fairfax Station (1991)
17. Geyer, C.J., Thompson, E.A.: Annealing Markov chain Monte Carlo with applications to ancestral inference. *J. Am. Stat. Assoc.* **90**(431), 909–920 (1995)
18. Goetze, F.: Rate of convergence of simulated annealing processes. preprint (unpublished) (1992)
19. Häggström, O.: *Finite Markov Chains and Algorithmic Applications*. London Mathematical Society Student Texts. Cambridge University Press, Cambridge/Cambridge (2002)
20. Hajek, B.: Optimization by simulated annealing: a necessary and sufficient condition for convergence. In: *Adaptive statistical procedures and related topics* (Upton, N.Y., 1985), vol. 8 of *IMS Lecture Notes Monogr. Ser.*, pp. 417–427. Inst. Math. Statist., Hayward, CA (1986)
21. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**(1), 97–109 (1970)
22. Holley, R., Stroock, D.: Simulated annealing via Sobolev inequalities. *Commun. Math. Phys.* **115**(4), 553–569 (1988)
23. Horn, R.A., Johnson, C.R.: *Topics in Matrix Analysis*. Cambridge University Press, Cambridge (1991)
24. Jain, S.R., Gural, A., Wu, M., Dick, C.: Trained uniform quantization for accurate and efficient neural network inference on fixed-point hardware. arXiv preprint [arXiv:1903.08066](https://arxiv.org/abs/1903.08066), 6 (2019)
25. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)
26. Kou, S.C., Zhou, Q., Wong, W.H.: Equi-energy sampler with applications in statistical inference and statistical mechanics. *Ann. Statist.* **34**(4), 1581–1652 (2006). With discussions and a rejoinder by the authors
27. Löwe, M.: On the convergence of genetic algorithms. *Expos. Math.* **14**(4), 289–312 (1996)
28. Löwe, M.: Simulated annealing with time-dependent energy function via Sobolev inequalities. *Stoch. Process. Appl.* **63**(2), 221–233 (1996)
29. Löwe, M., Vermet, F.: The swapping algorithm for the Hopfield model with two patterns. *Stoch. Process. Appl.* **119**(10), 3471–3493 (2009)

30. Mac Culloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**, 115–133 (1943)
31. Madras, N., Zheng, Z.: On the swapping algorithm. *Random Struct. Algorithms* **22**(1), 66–97 (2003)
32. Marinari, E., Parisi, G.: Simulated tempering: a new Monte Carlo scheme. *Europhys. Lett.* **19**(6), 451–458 (1992)
33. Orlandini, E.: Monte carlo study of polymer systems by multiple markov chain method. In: *Numerical methods for polymeric systems*, Whittington, S.G. (ed)., IMA Volume in Mathematics and Its Applications, vol. 102, pp. 33–57. Springer, New York (1998)
34. Rosenblatt, F.: *Principles of Neurodynamics: Perceptron and Theory of Brains Mechanisms*. Spartan Books, Washington (1962)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.