

GROUP2505 – Log-likelihood analysis for Restricted Boltzmann Machines

Elias Maria Bonasera, Alberto Casellato, Nicola Garbin, and Francesco Pazzocco
(Dated: March 28, 2025)

Here, in the “abstract” (more or less of 8 lines), there should be a short summary of the work and of its main findings. In a paper, the abstract is important also for attracting potential readers, hence it is convenient to start it with some catchy statement. — The rest of this text has the double purpose of (a) providing a template for the assignment latex, and (b) introducing how to structure the backbone of the text and explaining some details of this assignment. The “zzz” fill the space to show a typical (but not rigid) extension of the parts.

ZZZZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ ZZZZZZZ
Z ZZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ

1. INTRODUCTION

The main goal of this assignment is to explore how the performance of an RBM changes for different choices of the hyperparameters of the model, using the MNIST digits as the database; in particular, using the log-likelihood evaluation, we explore the trend of the model during the learning process as the number of epochs increases.

In section 2. it is exposed a brief insight of the Restricted Boltzmann Machines architecture. The log-likelihood is introduced in this section as it is adopted both to estimate the performance of the model and as an evaluation of $\nabla_{\theta} E(\theta)$ in optimizers (see Eq. 8 and Eq. 9). Subsection 3.1. is opened with the explicit expression for the partition function Z in the Bernoulli case (Eq. 5) and then it regards theoretical insight about setting parameters of the model. In subsection 3.2. operative choices regarding simulations are discussed, including numerical methods adopted to prevent computation errors. The explicit expression for the partition function Z in the Spin case (Eq. 6) is introduced in this subsection, since it is approximated in a similar way compared to the Bernoulli case. In section 4. results concerning the log-likelihood are discussed, dependent on number L of hidden units (Figure 4), CD steps number (Figure 5), γ parameter (..), learning rate (..), optimizers choice (..), Bernoulli vs Spin case (..), POTTS on/off for SGD optimizer (..) and POTTS on/off for RMSprop optimizer (..).

2. RESTRICTED BOLTZMANN MACHINES

Restricted Boltzmann Machines (RBM) are a powerful kind of generative models designed to accomplish training processes relatively fast. In RBMs, a set of D binary visible units i of state $\{v_i\}_{i=1,\dots,D}$ is symmetrically connected to a set of L binary hidden units μ of state $\{h_\mu\}_{\mu=1,\dots,L}$; the continuous weight $w_{i\mu}$ quantifies the strength between unit i and unit μ (see Figure 1). RBMs use an energy function to define the probability distribution over the input data [1] [2].

In the training process the energy of the configuration is minimized by adjusting the parameters θ . One of

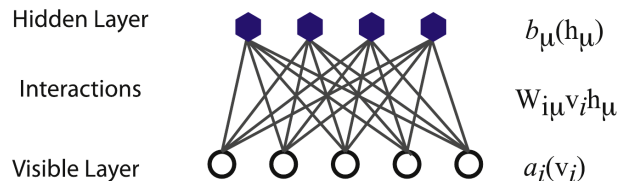


FIG. 1. A Restricted Boltzmann Machine (RBM) is made up of visible units, denoted as v_i , and hidden units, represented as h_μ . These units engage with one another through interactions characterized by the weights $w_{i\mu}$. Notably, there are no direct interactions among the visible units or among the hidden units themselves.

the most adopted training algorithm is Contrastive Divergence which allows to approximate the gradient of the likelihood to update the parameters. During this process, a cyclic Gibbs sampling is performed setting the visible units given the hidden ones and vice versa, according to the following probabilities p :

$$p(h_\mu = 1 \mid \mathbf{v}) = \sigma(b_\mu + \sum_i v_i w_{i\mu}) \quad (1)$$

$$p(v_i = 1 \mid \mathbf{h}) = \sigma(a_i + \sum_{\mu} h_{\mu} w_{i\mu}) \quad (2)$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic sigmoid function, a_i is the bias of the i -th visible unit and b_μ is the bias of the μ -th hidden unit; they act shifting the sigmoid function $\sigma(x)$. The absence of links among units of the same type simplifies the training process. Moreover, the number of iterations of Eq. 1 and Eq. 2 can be set to 2 if real data is used to fix $\{v_i\}_{i=1,\dots,D}$ in the first place.

The goodness of a model is evaluated by computing the log-likelihood \mathcal{L} of the data:

$$\mathcal{L} = \frac{1}{M} \sum_{m=1}^M \ell_{\theta}(v^{(m)}) \quad (3)$$

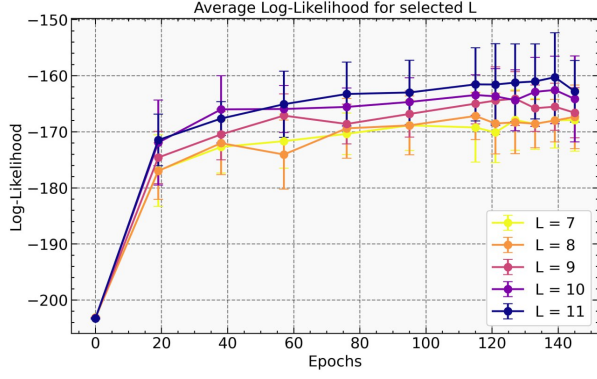


FIG. 2. The evolution of the log-likelihood over training epochs for each L is shown. The curves represent the average log-likelihood across 10 stochastically distinct runs.

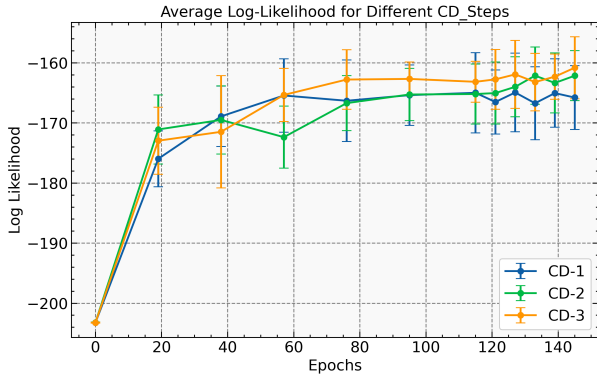


FIG. 3. The figure shows the evolution of the log-likelihood over training epochs for different numbers of CD steps. The curves represent the average log-likelihood across 10 stochastically distinct runs.

$$\ell_{\theta}(v^{(m)}) = \log \sum_h e^{-E(v,h)} - \log Z \quad (4)$$

where M is the number of data points, Z is the partition function, $E(v,h)$ in the exponential argument $e^{-E(v,h)} = G(h) \prod_i e^{H_i(h)v_i}$ is the energy function, with $G(h) = \prod_{\mu} e^{b_{\mu}h_{\mu}}$ and $H_i(h) = a_i + \sum_{\mu} w_{i\mu}h_{\mu}$, and, in Eq. 4, the index h of the summation represents each possible state of the hidden layer.

3. METHODS

3.1. Theoretical implementation

Regarding the computation of the partition function Z (in Eq. 4), its implementation depends on the set of values which each unit can be set on. Since the direct computation of the Z function is unfeasible, due to the summation of all possible state combinations of the system, a possible approach is to split the summation separately over states v and h .

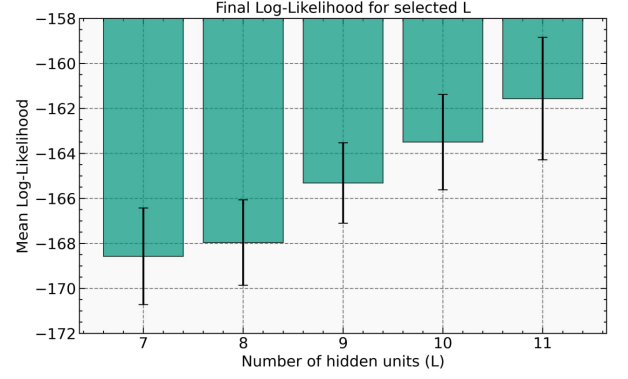


FIG. 4. Bar chart displaying the weighted mean and standard deviation of the log-likelihood over the last epochs for different values of L .

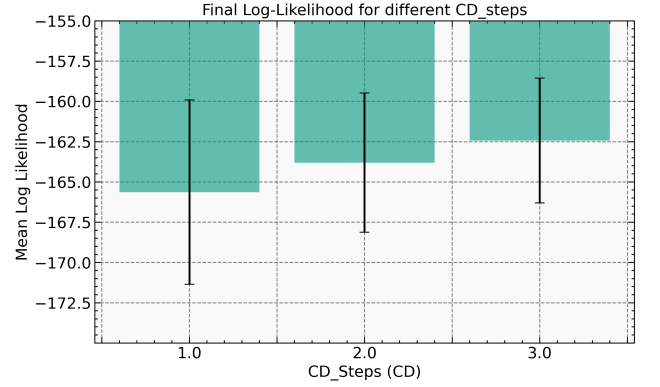


FIG. 5. Bar chart displaying the weighted mean and standard deviation of the log-likelihood over the last epochs for different numbers of CD steps.

For the Bernulli case the units can assume the values $\{0,1\}$ and the partition function form can be manipulated to obtain the following expression:

$$Z = \sum_v \sum_h e^{-E(v,h)} = q^D \sum_h G(h) \prod_{i=1}^D \frac{1 + e^{H_i(h)}}{q} \quad (5)$$

where q is the average of $1 + e^{H_i(h)}$ $\forall i, h$ that we introduce to avoid overflow. Concerning the training process of RBM, v_i are set using real data. Weights $w_{i\mu}$ are initialized sampling values from a Gaussian distribution of mean 0 and standard deviation of $2/\sqrt{L+D}$. Biases b_{μ} of hidden units are initialized to 0. Thus h_{μ} are evaluated by Eq. 1 before passed to Eq. 2 in order to compute back v_i . In such evaluation it is convenient to set $a_i = \log[t_i/(1-t_i)]$; fixed the i -th visible unit, t_i is defined as the number of times such unit is 1 over the whole training array set, normalized over the size of the set. Referring to Eq. 2, the choice of shifting the sigmoid $\sigma(x)$ by the function $\log[t_i/(1-t_i)]$ of the average t_i ensures that the units can activate properly, even given initial weights $w_{i\mu}$ close to 0.

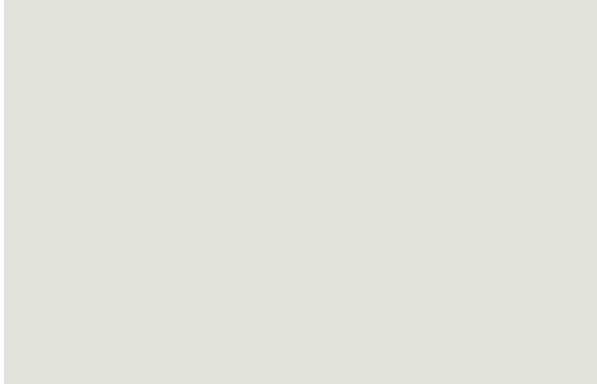


FIG. 6. Description...

The initialization of biases a_i , firstly proposed by Hinton [2], makes hidden units able to activate and differentiate better their activation based on data, avoiding the risk of getting stuck on values near to 0.

3.2. Computation

We define two Python3 classes, one for the computation of the log-likelihood and the other for defining the RBM structure and managing all the input and output of the training process. Referring to Eq. 4, in order to address the problem of overflow of $e^{H_i(h)v_i}$, we approximate the summation of powers over h as a sum of the biggest ones. The purpose of this choice is to extract the biggest term out of the summation thus the applied logarithm avoids exponentiation.

Regarding the expression of the partition function shown in Eq. 5, it changes form in Spin case, since it is characterized by units taking values from $\{-1, 1\}$; manipulating the second member of the equation one gets:

$$Z = \sum_v \sum_h e^{-E(v,h)} = \sum_h G(h) \prod_{i=1}^D 2 \cosh(H_i(h)) \quad (6)$$

For Eq. 6 the $2 \cosh(H_i(h))$ term is approximated to just $e^{H_i(h)}$ by adding a common positive value to all the exponents, which shifts them away from negative domain, and compensating that out of the summation. Thus the same approximation procedure described for Eq. 4 is applied to Eq. 6.

To upload the values of the weights and biases during the training phase, we adopted two different gradient descent types: SGD and RMSprop, whose update equations are found in the Appendix (Eq. 8 and Eq. 9); we adopt the versions with the “+” sign, for gradient ascent, since we aim to maximize a monotonically increasing function. The LASSO regularization (Eq. 7), controlled by the constant γ , is performed after applying gradient ascent during training.

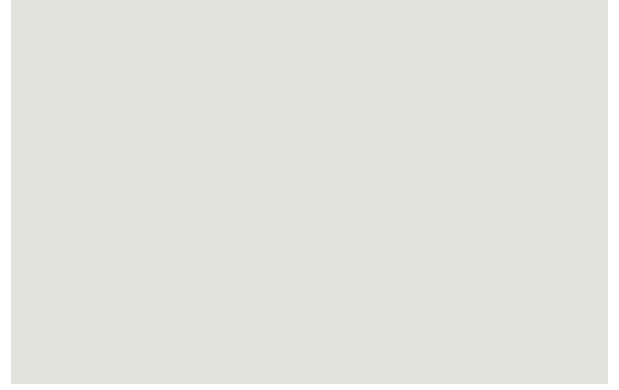


FIG. 7. Description...

We define a function to visualize the evolution of the log-likelihood of a specific model throughout the training phase and another function to compute and display the mean log-likelihood and standard deviation over the last few epochs. For our analysis, we select the set of digits $\{0, 1, 2\}$ from the MNIST database. We then evaluate the performance of the RBM by utilizing the visualization functions to compare different configurations of hyperparameters. In order to account for stochasticity, for each chosen set of hyperparameters we first perform ten independent training runs varying the NumPy random seed and then compute the mean log-likelihood accross runs obtaining an average trend. To derive a final log-likelihood value for each hyperparameter configuration, we calculate a weighted average of the values of the log-likelihood over the last few training epochs. Specifically, we examine the influence of variations in the number of Contrastive Divergence steps (CD), in the number L of hidden units and in parameters such as the learning rate η , ϵ and γ ; here ϵ is a coefficient in the RMSprop update rule (see Eq. 9).

Additionally, we analyze the impact of the optimizer choice for models with one-hot encoding enabled (POTTS) or disabled and either Bernoulli or Spin variables.

4. RESULTS

Describe what you found.

Cite Figure ??(a), etc. to add information. Later also cite Figure 6 and Figure 7. Of course the number and size of figures may vary from project to project.

Cite Table ?? to collect useful data in a clear way.

zzzzzzzzzzzzzzzzzzzz zzzzzz zzzzzzzzzzzz zzzzzzzz z zzzzzzzzzzzz
 zzzzz zzzzzzzzzzzz zzzzzzzzzzzzzzzzzzzzz zzzzzz zzzzzzzzzzzz zzzzzzzz
 z zzzzzzzzzzzzzzz zzzzz zzzzzzzzzzzz zzzzzzzzzzzzzzzzzzz zzzzzz
 zzzzzzzzzzzz zzzzzzzzz z zzzzzzzzzzzzz zzzzz zzzzzzzzzzzz
 zzzzzzzzzzzzzzzzzzzzz zzzzzz zzzzzzzzzzzz zzzzzzzzz z zzzzzzzzzzzzz
 zzzzz zzzzzzzzzzzz zzzzzzzzzzzzzzzzzzzzz zzzzzz zzzzzzzzzzzz zzzzzzzzz z
 zzzzzzzzzzzzzzz zzzzz zzzzzzzzzzzz.

ZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ
Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ
ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZZ ZZZZ ZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ
Z ZZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ. ZZZZZZZZZZZZZZZZZ ZZZZZ
ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZZ ZZZZ ZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ
Z ZZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ
ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZZ ZZZZ ZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ.

ZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ
Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZZ
ZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ
ZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z
ZZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ.

ZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ
Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ
ZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZ
ZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z
ZZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ.

ZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ
Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZZZ
ZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ
Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZ. ZZZZZZZZZZZZZZZZ ZZZZZ
ZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZ
ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ
Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ
ZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZ
ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ
ZZZZ ZZZZZZZZZ.

[illegible]

ZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ
Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ
ZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ ZZZZ ZZZZZZZZZ
ZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z ZZZZZZZZZZZ
ZZZZ ZZZZZZZZZZ ZZZZZZZZZZZZZZZZZ ZZZZZ ZZZZZZZZZZZ ZZZZZZZ Z
ZZZZZZZZZZZZ ZZZZ ZZZZZZZZZZ.

5. CONCLUSIONS

Discuss the key aspects that we can take home from this work.

Check if your text is light, swift, and correct in exposing its passages.

APPENDIX

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \gamma \cdot \eta_t \cdot \text{sign}(\boldsymbol{\theta}) \quad (7)$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t \pm \eta_t \nabla_{\boldsymbol{\theta}} E(\boldsymbol{\theta}) \quad (8)$$

$$\begin{cases} \mathbf{g}_t = \nabla_{\theta} E(\theta) \\ \mathbf{s}_t = \beta \mathbf{s}_{t-1} + (1 - \beta) \mathbf{g}_t^2 \\ \theta_{t+1} = \theta_t \pm \eta_t \frac{\mathbf{g}_t}{\sqrt{\mathbf{s}_t + \epsilon}} \end{cases} \quad (9)$$

- [1] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, David J. Schwab, *A high-bias, low-variance introduction to Machine Learning for physicists*, Physics Reports **810**, 1–124 (2019).
- [2] Geoffrey E. Hinton, *A Practical Guide to Training Restricted Boltzmann Machines*, Neural Networks: Tricks of the Trade: Second Edition, 599–619 (2012).