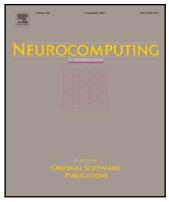
Neurocomputing 610 (2024) 128469

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Learning restricted Boltzmann machines with pattern induced weights

J. Garí [a], E. Romero [b], F. Mazzanti [a,*]

[a] Departament de Física, Campus Nord B4-B5, Universitat Politècnica de Catalunya, E-08034 Barcelona, Spain
[b] Departartment de Ciències de la Computació, Campus Nord Omega, Universitat Politècnica de Catalunya, E-08034, Barcelona, Spain

ARTICLE INFO

Communicated by K. Li

ABSTRACT

Restricted Boltzmann Machines are energy-based models capable of learning probability distributions. In practice, though, it is seriously limited by the fact that the computational cost associated with the exact evaluation of the gradients, required during learning, is prohibitively high. The standard approach to mitigate this problem is to use the Contrastive Divergence algorithm, but it leads to a rough approximation that presents issues on its own. As a completely different alternative, a model called RAPID (Pozas-Kerstjen et al., 2021) recently appeared, where unit weights are constructed from high-probability patterns that allow for an effective evaluation of the update rules along learning. In this work we analyze RAPID to find that it also presents some drawbacks that constrain its performance. We identify the problematic sources in RAPID and modify them accordingly to build a similar but more flexible alternative, called PIW (Pattern Induced Weights). Experiments show that PIW performs better than the original RAPID implementation, bringing it to a competitive level when compared to a standard RBM with CDk, with a substantial reduction in the number of training parameters.

1. Introduction

Restricted Boltzmann Machines (RBMs) [1], which emerged as a simplified version of the more complex original Boltzmann Machine [2], have attracted the interest of the scientific community in recent years due to their capability to solve probabilistic problems. Being an energy-based model, the RBM follows a Boltzmann probability distribution similar to the ones found in the Ising model or the spin glass magnetic systems. Once trained, RBMs can be used as generative stochastic neural networks that can learn a probability distribution over its input space [3].

While the original formulation of the RBM considers only binary units, different variants have been explored over the years, and are still the subject of current research. For instance Gaussian-Bernoulli RBMs [4–8], designed to handle continuous data, use visible and hidden units following a Gaussian and Bernoulli distributions, respectively. Furthermore, multinomial units have been successfully applied to treat categorical data [9,10]. Other models that focus on changing the nature of the visible and/or hidden units can also be found in the literature [11–13].

Restricted Boltzmann Machines have also been used as the main building blocks of relevant architectures such as Deep Belief Networks [14–16], Deep Autoencoders [17–19], Convolutional Restricted Boltzmann Machines [20,21] or Recurrent Temporal Restricted Boltzmann Machines [22,23]. All these models extend the capabilities of standard RBMs, fostering active research in many areas of general interest, such as, for example, image analysis and classification [17,24,25], collaborative filtering [9,26], acoustic modeling [27], or quantum physics [28–30]. Nowadays RBMs are still an active area of research [31–33], although they are not the only kind of neural network that can handle such problems, as the recent literature shows [34–36].

Despite the elegant formulation of the RBM neural network, in practice it presents a number of challenging issues that have prevented from its wide application to solve realistically large problems. The main drawback can be actually traced down to the evaluation of the normalization constant that enters the Boltzmann probability distribution, the Partition function. In fact, this problem is more sound than it appears as the learning update rule implies a statistical average over all possible system states in what is called *the negative phase*. This problem is usually circumvented using the Contrastive Divergence (CD) algorithm [37], which replaces the exact negative phase by a sum over a set of states generated performing k Gibbs steps starting from the training set. This procedure is usually denoted CDk.

While CDk has found its way in many applications, it is quite a rough approximation that is only guaranteed to be reliable in the $k \to \infty$ limit [38,39]. For that reason, alternative approximations are of relevance [40–42]. In this way, as an attempt to bypass this problem, RAPID (Regularized Axons Pattern InDuced correlations) [43] was developed. Using patterns as trainable parameters for Hopfield-like weights built from the Hebb rule, the authors of RAPID created their

* Corresponding author.
E-mail address: ferran.mazzanti@upc.edu (F. Mazzanti).

https://doi.org/10.1016/j.neucom.2024.128469
Received 29 May 2023; Received in revised form 3 June 2024; Accepted 25 August 2024
Available online 28 August 2024

model inspired on the physics of magnetic spin systems, where privileged spin configurations have low energies and govern the behavior of the system. In its own nature, RAPID tries to mimic that, finding the configurations that make the training set have high probability. Additionally, in RAPID the negative phase is approximated by a simple expression that only depends on the patterns.

Anyway, the relation between the Hopfield model and the RBM is not new, and has been widely explored in the last years. For instance in [44], the thermodynamical equivalence between Hopfield networks and visible units in Bernoulli-Gaussian RBMs is demonstrated for uncorrelated patterns. In this work, weights connecting visible and hidden units are directly the patterns stored in the Hopfield network. These results are extended in [45] (see also [46] and references therein) with the same architecture, and in [47] with an additional set of hidden units. In [48] correlated patterns are considered, and the equivalence between Hopfield networks and Bernoulli-Gaussian RBMs is obtained when the weights of the latter are the Q_i vectors of the QR decomposition of the patterns matrix. In practice, this procedure can be used as a useful weights initialization in a standard RBM training. Other interesting works relating patterns and weights can be found in [49–51].

In this context, the novelty of RAPID is to use Hebb's rule to directly construct the weights of binary-binary RBMs, assuming these will be the most probable configurations during learning and once the network is trained. In this work, we analyze the behavior of RAPID using different metrics to find that it presents some drawbacks that limits its performance. With the aim to solve these issues, we modify the original RAPID model to build a new one that is more flexible and that is competitive when compared with a standard RBM and uses less trainable parameters. We call this model Pattern Induced Weights (PIW) RBM. The main ingredients of PIW are the use of patterns as the atomic quantities that determine the weights, and to restore CDk to compute the negative phase.

2. Restricted Boltzmann machines

The energy function of a binary RBM with N_v visible units $\mathbf{x}$ and N_h hidden units $\mathbf{h}$, is defined as [1,52]:

$$E(\mathbf{x},\mathbf{h}) = -\mathbf{x}^T\mathbf{W}\mathbf{h} - \mathbf{x}^T\mathbf{b} - \mathbf{c}^T\mathbf{h} \tag{1}$$

where $\mathbf{W}$ is the matrix of two-body weights connecting visible and hidden units, while $\mathbf{b}$ and $\mathbf{c}$ are the visible and the hidden bias terms, respectively. This energy defines the (Boltzmann) probability distribution described by the RBM, namely

$$p(\mathbf{x},\mathbf{h}) = \frac{e^{-E(\mathbf{x},\mathbf{h})}}{Z}\ , \tag{2}$$

with Z the normalization constant, usually called the partition function:

$$Z = \sum_{\mathbf{x},\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}\ . \tag{3}$$

In order to estimate the model parameters $\theta = \{\mathbf{W},\mathbf{b},\mathbf{c}\}$ one usually minimizes the Kullback–Leibler divergence (KL) between the empirical probability distribution of the training set, $q(\mathbf{x})$, and that of the RBM model, $p(\mathbf{x}\,|\,\theta)$.

$$\mathrm{KL}\left[q(\mathbf{x})\parallel p(\mathbf{x}|\theta)\right] = \sum_{\mathbf{x}\in S} q(\mathbf{x})\ln\frac{q(\mathbf{x})}{p(\mathbf{x}|\theta)} = -\ln(N_T) - \frac{1}{N_T}\sum_{i=1}^{N_T}\ln p(\mathbf{x}_i|\theta)\ , \tag{4}$$

where $S = \{\mathbf{x}_1,\dots,\mathbf{x}_{N_T}\}$ is the training set containing N_T elements, and $q(\mathbf{x}) = 1/N_T\ \ \forall\mathbf{x}\in S$. That is equivalent to maximize the log-likelihood of the training data with respect to the RBM model

For a generic training parameter θ, the KL is usually optimized using gradient descent, which leads to the standard equations [53]

$$-\frac{\partial(\mathrm{KL})}{\partial\theta} = \frac{1}{N_T}\sum_{\mathbf{x}_s\in S}\frac{\partial\mathcal{F}(\mathbf{x}_s)}{\partial\theta} - \sum_{\mathbf{x}} p(\mathbf{x}|\theta)\frac{\partial\mathcal{F}(\mathbf{x})}{\partial\theta} \tag{5}$$

in terms of the Free energy

$$\mathcal{F}(\mathbf{x}) = -\ln\left[\sum_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}\right]\ , \tag{6}$$

which in turn define the Boltzmann probabilities through the expression

$$p(\mathbf{x}) = \sum_{\mathbf{h}}\frac{e^{-E(\mathbf{x},\mathbf{h})}}{Z} \equiv \frac{e^{-\mathcal{F}(\mathbf{x})}}{\sum_{\mathbf{z}} e^{-\mathcal{F}(\mathbf{z})}}\ . \tag{7}$$

The complexity associated to minimizing the KL lies in the computation of the second term in the rhs of Eq. (5), usually called the *negative phase*, needed to compute the update rule for the parameters θ. The negative phase is typically approximated using CD [37], which eliminates the forbidding task of evaluating the partition function Z.

3. RAPID

Due to the complex dynamics implied in standard RBM training, the authors of Ref. [43] proposed a new method designed to improve convergence and to reduce the computational cost. This model, denoted as RAPID, is inspired on the thermodynamics of spin glass systems, bringing some of the well-established statistical mechanics knowledge acquired along the years to the world of RBMs [54,55].

As is well known, a generic Boltzmann Machine initialized with Gaussian random weights behaves as the Sherrington and Kirkpatrick spin glass model [56,57]. In this scenario, the system presents a strong degree of frustration and the set of lowest energy states, which dominate the Boltzmann probability distribution, is both very large and very difficult to determine. In this way, finding this set becomes a NP-complete problem.

In fact, the authors of Ref. [43] argue that a properly trained RBM should never enter a spin glass phase. This is because, in contrast to what happens in a spin glass, only a limited number of states is expected to acquire a significant amount of probability mass. Based on that, a modification of the standard RBM model, called Regularized Axons (RA), is proposed in [43] to avoid this problem.

In RA, the weights are built from a number K of patterns $\{\vec{\xi}^{(k)}\}$, through the Hebb rule [58] as in the Hopfield model [59]

$$w_{i\alpha} = \frac{1}{\sqrt{K}}\sum_{k=1}^{K}\xi_i^{(k)}\xi_\alpha^{(k)}, \tag{8}$$

where $\xi_i^{(k)},\ \xi_\alpha^{(k)} \in \{-1,+1\}$, with $k = 1,\dots,K$, $i = 1,\dots,N_v$ and $\alpha = 1,\dots,N_h$. Here we use Roman and Greek indexes to denote visible and hidden units, respectively. The set of patterns $\{\xi_i^{(k)},\ \xi_\alpha^{(k)}\}$ become then the fundamental quantities of the model and constitute the only trainable parameters. In its original form, RAPID has no bias terms. In contrast to the original Hopfield model, RAPID is not a standard associative memory since the $\{\vec{\xi}^{(k)}\}$ patterns do not represent memorized data, but are learnt along RBM training.

In the Hopfield model and according to [57], when $K \ll N$, the patterns are low-energy configurations because cross-talk terms are expected to be small. This suggests a new approximation to estimate statistical averages, called *Pattern-InDuced correlations* (PID) in [43], of the form

$$\langle f(\vec{\sigma})\rangle \approx \frac{1}{K}\sum_{k=1}^{K} f(\vec{\xi}^{(k)}), \tag{9}$$

for any function of the units $f(\vec{\sigma})$. In this approximation one assumes that the K patterns $\{\vec{\xi}^{(k)}\}$ exhaust the complete set of low energy states and have a similar energy. Furthermore, for Eq. (9) to be a good approximation, it is assumed that the energy of the patterns is much lower than the energy of any other state, which can apply or not depending on the problem at hand. In any case, Eq. (9) allows for a fast approximation of the negative phase, and is fully adopted in RAPID. Finally, notice that in order to guarantee that the $\{\vec{\xi}^{(k)}\}$ patterns represent physical spin configurations, they must be binary.

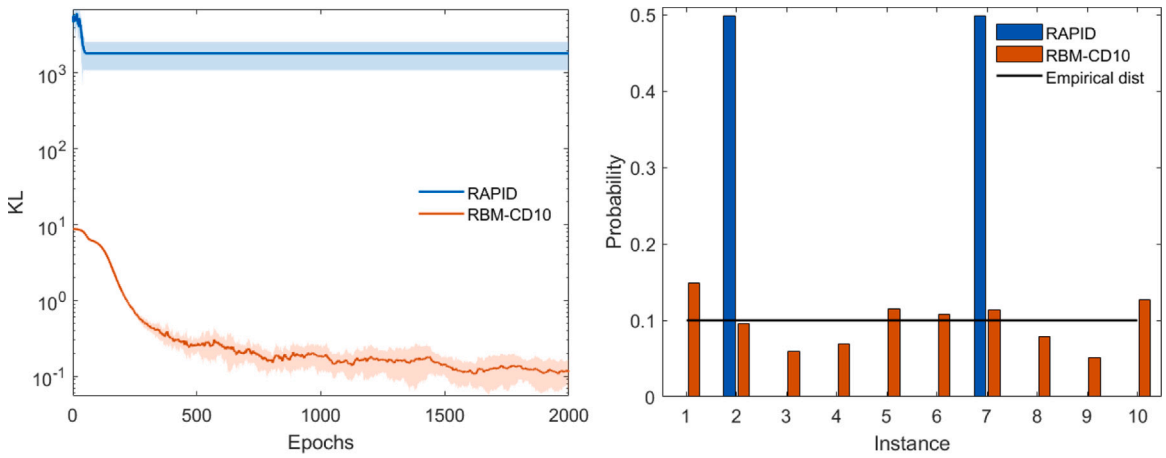


Fig. 1. Evolution of the KL along learning (left) and final probabilities of the training set (right) for both the RAPID (blue) and the standard RBM (red) models. The shaded areas in the left panel indicate the standard deviation obtained in different repetitions of the experiments. In the right panel, the final probabilities are compared with the empirical distribution (black).

Training a RAPID model, as in standard RBMs, consists in looking for the optimal values of the patterns $\{\xi_i^{(k)}, \xi_\alpha^{(k)}\}$ which minimize the KL with respect to the empirical probability distribution associated to a training dataset $\mathcal{S}$. This is performed using a gradient ascent scheme with Eq. (5), computing the positive phase exactly and the negative phase approximated with PID [43].

4. Testing RAPID experimentally

In this section we check the performance of the RAPID model in different experiments. We reproduce the results on simple problems originally presented in [43], and extend the analysis to other quality estimators.

As in [43], we use the 4 × 4 Bars dataset, consisting on 4 × 4 pixel images of vertical bars with ±1 values. In this way, one can build a total of 16 different Bar images, which include one with no bars (with all pixels in white), and another with four bars (with all pixels in black). However and in order to establish a fair comparison, we follow the procedure used in [43] where these two last examples are not considered. Therefore, the 4 × 4 Bars dataset contains 14 examples, split in 10 training and 4 test images, respectively. In this sense, this dataset is a toy model where all calculations are easily done, including complex estimators that are out of reach in real size problems. As in [43], we train a RBM with RAPID consisting of 16 visible and 1000 hidden units, and K = 8 patterns. For the sake of comparison, we also train a standard CD10 RBM with 80 hidden units. This RBM has a significantly smaller number of trainable parameters, but has been chosen so as to follow the standard procedure of having a number of hidden units equal to a small multiple of the number of visible ones. In order to collect enough statistics, we have performed 20 independent training runs for both (RAPID and standard RBM) models.

Restricted Boltzmann Machines with a small number of visible units allow for the exact computation of the probabilities in Eq. (2), which in turn grants access to the KL. This is the case of the 4 × 4 Bars dataset. The authors in [43] analyze the Hamming Distance (HD) of the reconstructed images to the training set instead. In their analysis they conclude that RAPID converges faster than a standard RBM to a low HD value. We have been able to reproduce this behavior in our experiments. However, the cost function being optimized during learning is not the HD but the KL, so this is the main quantity we will address in the following discussion.

In order to better clarify this last statement, we compute the KL divergence along learning for both RAPID and the standard RBM. We also evaluate the model probability distributions after learning, and compare it with the empirical probability distribution of the data. The left panel of Fig. 1 shows the evolution of the KL, while the resulting probabilities of the training examples are depicted in the right panel (blue and red bars for RAPID and standard RBM, respectively). The results in the left panel show the mean and standard deviation of 20 independent runs, while the right plot depicts the resulting probabilities for one typical run. The first and most relevant conclusion one can draw from the figure is that RAPID is unable to optimize the learning KL cost function, reaching final values orders of magnitude worse than those of the standard RBM. This is very surprising considering the small size of the problem at hand. A direct consequence is that the probabilities assigned to the training set differ significantly from the empirical ones, which are uniform and equal to 0.1 (black line in the right panel). In most of our experiment repetitions, RAPID assigns almost one hundred percent of the probability mass to just two states of the training set, leaving almost nothing for all other vectors of the space. Additionally, these two states are the reverse of each other, consistently with the absence of bias terms that help breaking this symmetry. Notice, though, that the extremely large KL values reported are a consequence of the fact that the small probabilities assigned to the training vectors are *really* small, while in practical terms it is irrelevant whether a resulting probability is 10^{-10} or 10^{-100}, the latter leading to improperly large value of the KL. On the other hand, the standard RBM produce more balanced probabilities, consistent with much lower KL values.

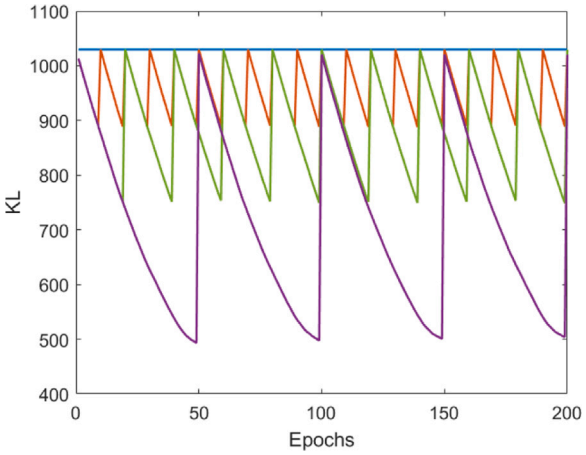


Fig. 2. KL for RAPID along learning with different binarization periods $N_{bin} = 1, 10, 20, 50$ in the 4 × 4 Bars dataset.

Another important ingredient of the RAPID model is to use binarized patterns. Of course, the use of standard learning techniques breaks

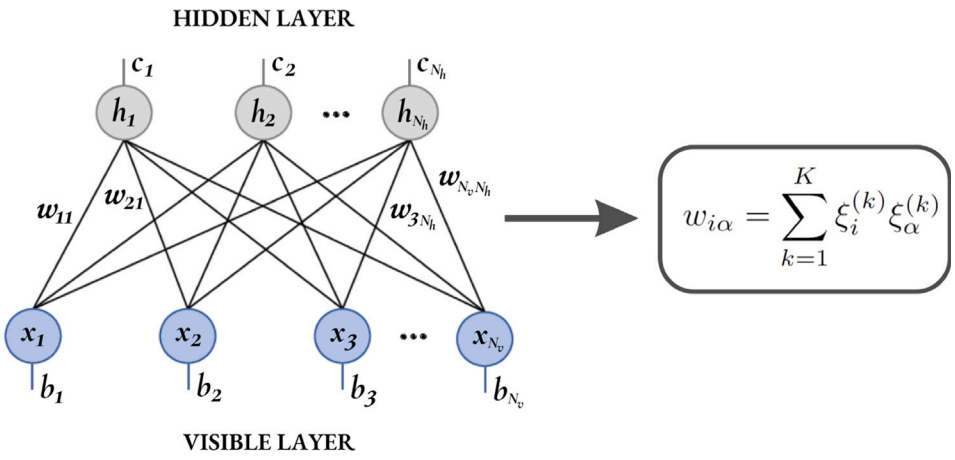


Fig. 3. The PIW network architecture.

this property at each epoch, so the authors binarize their values along learning. However, this is a delicate procedure as the probabilities assigned by the RBM to a given state change dramatically after this procedure. This is a direct consequence of the fact that the energy is a polynomial in the patterns (through the weights) while the joint (visible and hidden) probabilities are proportional to the exponential of the energy. The impact of this is shown in Fig. 2, which shows the evolution of the KL while learning, at constant learning rate. The different curves show the result produced by RAPID changing the interval N_{bin} at which the binarization is performed. As can be seen, this produces notorious spikes that reset the KL to undesired large values. This is probably the reason why the authors in [43] use an uncommonly large decreasing learning rate scheme, which enforces convergence by itself.

5. Pattern induced weights RBM

The previous analysis show that the main ingredients in the RAPID model lead to large values of the KL loss function even in small sized problems where a standard RBM performs better. For that reason we look for suitable modifications that improve the original RAPID model. We denote by PIW (Pattern Induced Weights RBM) the resulting model. Despite the modifications described below, the topology of the network is very similar to the one used in RAPID, as shown in Fig. 3. The PIW network is a RBM that shares with RAPID the fact that weights are no longer atomic items but rather computed from the fundamental quantities of the model, which are the set of $\{\vec{\xi}^{(k)}\}$ patterns. As discussed below, the PIW model incorporates bias terms, which are missing in the original RAPID network.

The first and more evident change, according to the results shown in Fig. 2, is to discard the binarization of the $\vec{\xi}^{(k)}$ during learning. This implies that the patterns can no longer be understood as spin configurations. However, the fact that the patterns can take real values induces plasticity into the model, allowing it to adapt better to the problem at hand. Notice that the binarization, together with the definition of RA in Eq. (8) imply that the weights are constrained to be in a bounded interval, which can be of relevance since bounded weights imply bounded energies which in turn imply bounded probabilities. Furthermore, the original definition of RA in Eq. (8) leads to much larger weights than what one usually find in standard RBMs. Although this scenario typically avoids the problems related to having the system in a spin glass phase, it produces largely unbalanced probabilities and therefore large KL values, as shown above. Anyway and as in RA, we still keep real valued, non-binary patterns as the fundamental quantities defining our model. We initialize them following a Gaussian distribution centered at the origin, normalized to 1. Once learning starts, no further normalization is performed afterwards.

The second change, which is in fact related to the previous one, is to remove the $1/\sqrt{K}$ constant from the RA scheme of Eq. (8), leading to

$$w_{i\alpha} = \sum_{k=1}^{K} \xi_i^{(k)} \xi_\alpha^{(k)}. \tag{10}$$

Since we do not binarize the patterns but let them take any real value during learning, an overall multiplicative factor in the weights becomes unnecessary. This removes any a priori constraint on the $\omega_{i\alpha}$ weights, letting the learning process decide what is the proper norm the patterns should take.

The modification of RA has a direct impact on PID, the other main idea of RAPID. As previously mentioned, PID assumes that statistical averages can be approximated using only the set of K patterns $\{\vec{\xi}^{(k)}\}$. As in the Hopfield model, though, this picture is broken when cross-talk terms become relevant, something that cannot be known *a priori*. Besides that, since patterns are no longer binary in PIW, the PID approximation loses its physical insight. We therefore discard the PID approximation, replacing the evaluation of the negative phase with a standard Contrastive Divergence approximation. To that end, the CDk algorithm has been adapted accordingly to be applied directly to the $\vec{\xi}^{(k)}$ patterns rather than to the RBM weights and biases as is usually done. Notice also that, once the PID approximation is discarded, there is no need to constrain the number of patterns to be much smaller than the total number of units.

We also include biases into the model, a feature that is missing in the original RAPID formulation, meant to break the parity symmetry of the energy function. There are two immediate ways to incorporate biases into the model: to consider them as independent new quantities, or to make them dependent on the patterns. In the end and provided the number of components of each pattern is large, this does not make a big difference. In this work we have adopted the first option, and modify them along learning with standard CDk. With all these modifications the free energy reads, considering $\{-1,+1\}$ binary states

$$\mathcal{F}(\mathbf{x}_s) = -\sum_{\alpha=1}^{n} \ln\left[2\cosh\left(\sum_{i=1}^{m} w_{i\alpha} x_{si} + c_\alpha\right)\right] - \sum_{i=1}^{m} b_i x_{si}\ , \tag{11}$$

while the gradients with respect to the visible and hidden sectors of the patterns become, respectively

$$\frac{\partial \mathcal{F}(\mathbf{x}_s)}{\partial \xi_i^{(k)}} = -x_{si} \sum_{\alpha=1}^{n} \xi_\alpha^{(k)} \tanh\left(\sum_{j=1}^{m} w_{j\alpha} x_{sj} + c_\alpha\right), \tag{12}$$

$$\frac{\partial \mathcal{F}(\mathbf{x}_s)}{\partial \xi_\alpha^{(k)}} = -\sum_{i=1}^{m} \xi_i^{(k)} x_{si} \tanh\left(\sum_{j=1}^{m} w_{j\alpha} x_{sj} + c_\alpha\right)\ . \tag{13}$$

In much the same way, the gradient with respect to the visible and hidden biases read

$$\frac{\partial \mathcal{F}(\mathbf{x}_s)}{\partial b_i} = -x_{si} \ , \tag{14}$$

and

$$\frac{\partial \mathcal{F}(\mathbf{x}_s)}{\partial c_\alpha} = -\tanh\left(\sum_{j=1}^{m} w_{j\alpha} x_{sj} + c_\alpha\right) \ . \tag{15}$$

These equations are the basic ingredients of PIW learning, which follows the standard structure of a stochastic gradient ascent procedure with CDk, as summarized in Algorithm 1. First, patterns are randomly initialized from a low variance Gaussian distribution centered at the origin. Then, for every epoch and every minibatch, CDk samples are computed and their derivatives estimated using Eqs. (12)–(15). Finally, patterns and biases are updated and $w_{i\alpha}$ recomputed. In our experiments we used an exponentially decreasing learning rate scheme, which is also a common scheme when training standard RBMs. All these modifications seek to mitigate the issues found in Section 4, leading to a more flexible model when compared to the original RAPID one.

Algorithm 1 Training procedure for PIW

Input: dataset X
number of patterns K
number of hidden units H
number of minibatches N_B
number of epochs N_E
number of Gibbs steps k_G
learning rate λ
momentum parameter ν
Output: trained weights and patterns W, b, c, ξ

Initialize $\theta = \{\xi, b, c\}$ according to K and H
Compute $w_{i\alpha} = \sum_{k=1}^{K} \xi_i^{(k)} \xi_\alpha^{(k)}$
for $m \leftarrow 1$ to N_E **do** ▷ Loop over epochs
  for $n \leftarrow 1$ to N_B **do** ▷ Loop over minibatches
    Select the n-th minibatch $x_n^{(0)} \in X$
    for $t \leftarrow 0$ to $k_G - 1$ **do** ▷ Gibbs sampling
      Sample $h^{(t)} \sim p(h \mid x_n^{(t)})$
      Sample $x_n^{(t+1)} \sim p(x \mid h^{(t)})$
    $d\mathcal{F}_\theta \leftarrow \frac{\partial \mathcal{F}(x_n^{(0)})}{\partial \theta} - \frac{\partial \mathcal{F}(x_n^{(k_G)})}{\partial \theta}$ ▷ CDk using Eqs. (12)-(15)
    $\Delta\theta^{(m)} \leftarrow \nu \cdot \Delta\theta^{(m-1)} + \lambda \cdot d\mathcal{F}_\theta$
    $\theta \leftarrow \theta + \Delta\theta^{(m)}$ ▷ Update patterns and biases
    Compute $w_{i\alpha} = \sum_{k=1}^{K} \xi_i^{(k)} \xi_\alpha^{(k)}$

6. Experimental results

In this section, we analyze the performance of the PIW model in several different binarized datasets: the 4 × 4 Bars & Stripes [60], the MNIST [61], the Connect-4 [62] and the OCR-Letters [63]. For the sake of completeness, we compare the results with the ones obtained from a RBM trained with the standard Contrastive Divergence algorithm, and with the RAPID network.

6.1. 4 × 4 bars & stripes

Inspired by the 4 × 4 Bars analyzed by the authors of RAPID, we analyze the more standard 4 × 4 Bars & Stripes problem [64]. This is a small dataset of 30 images containing all possible combinations of vertical bars and horizontal stripes in 4 × 4 pixel images. We use all these states to train our models, so there is no test set to check against. In this case there are 16 visible units corresponding to each pixel in the input images, while the number of hidden units and patterns has to be properly selected. In the present case we have decided to use 60 hidden units in the standard RBM, which is large enough compared with the number of visible ones and corresponds to a total of 1036 training parameters (TP) considering biases and two-body weights. For the PIW one has to decide both the number of hidden units and patterns to include, as discussed in the following.

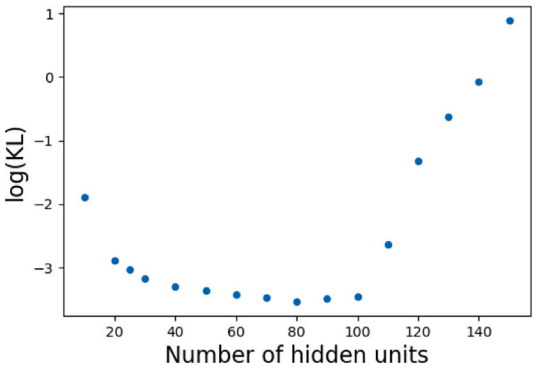


Fig. 4. Logarithm of the KL obtained at the end of training different PIW models using CD20 and a variable number of hidden units and patterns for the 4 × 4 Bars & Stripes dataset. In all cases the total number of training parameters is as close as possible to 1036.

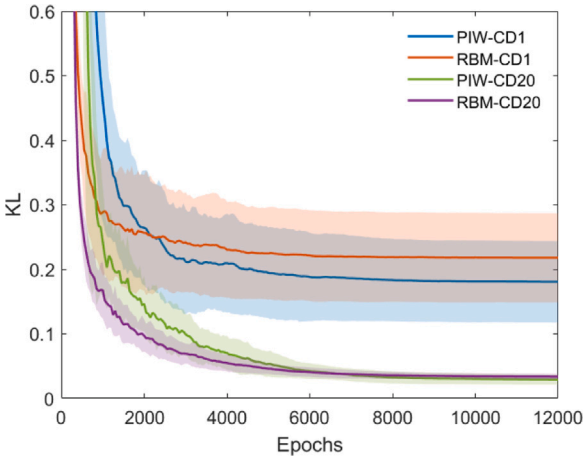


Fig. 5. Evolution of the KL for both the RBM and the PIW trained with CD1 and CD20 for the 4 × 4 Bars & Stripes dataset. The blue, red, green and magenta lines correspond to PIW with CD1, the standard RBM with CD1, PIW with CD20 and the standard RBM with CD20, respectively. The shaded regions indicate the standard deviation corresponding to 100 repetitions of each experiment.

In order to establish a suitable comparison between the RBM, the RAPID and the PIW models, we stick to the criteria of having the most similar amount of TP. In PIW this can be obtained from different combinations of number of patterns and hidden units, while the resulting models can perform differently. With no a priori knowledge of the optimal combination we have performed a model selection based on the lowest KL obtained in different repetitions of learning instances of the problem. The results obtained using CD20 are shown in Fig. 4. As it can be seen, there is a region of optimal configurations where the best KL changes only slightly. We have found that, for this particular dataset, the lowest KL corresponds to 80 hidden units and 10 patterns. In all cases, we have run 100 independent training simulations of 12 000 epochs each.

For the sake of completeness, we have also tested a larger standard RBM with 120 hidden units and a larger PIW network, both containing a similar amount of TP, which is close to twice the number in the smaller models. Additionally, we have also trained the original RAPID network with the same configurations. After the same model selection procedure performed for the small network, the resulting number of patterns and hidden units is $K = 26$ and $N_h = 60$. The final KL values corresponding to the different models averaged over 100 repetitions

Table 1
KL values for the 4 × 4 Bars & Stripes dataset with the different models and number of training parameters (in parenthesis).

Dataset	RBM CD1	PIW CD1	RBM CD20	PIW CD20	RAPID
4 × 4 BS	0.218 (1036)	0.181 (1056)	0.0336 (1036)	0.0291 (1056)	21.84 (1056)
4 × 4 BS	0.170 (2056)	0.161 (2052)	0.0301 (2056)	0.0280 (2052)	8.26 (2052)

are reported in Table 1, where the selected models have been trained using CD1 and CD20. Notice that RAPID does not use CD at any stage in its original formulation for such small problems. As it can be seen, the resulting KL of PIW are quite consistent and do not change significantly when doubling the number of TP of the model. In contrast, the performance of RAPID is not competitive with the other results, even though the KL values reported in the table are much lower than those shown in Fig. 1. This is due to the overwhelming difference in the number of hidden units employed in each case.

Fig. 5 shows the evolution of the KL along learning for the small models considered in Table 1. RAPID has not been included because of the large scale difference. In all cases we use CD1 and CD20 for the evaluation of the negative phase. As it can be seen from the figure, the final value of the CD20 curves are similar, while the curves in the CD1 case show a better performance of the PIW model. As expected, using 20 Gibbs steps leads to clearly better results, regardless of the choice of the model, while PIW performs slightly better in both cases.

6.2. MNIST

The binarized MNIST dataset [61] has 50 000, 10 000 and 10 000 training, validation and test vectors corresponding to 28 × 28 images of handwritten digits. This dataset, which is commonly used as a standard benchmark in Machine Learning, provides also the labels indicating which digit corresponds to each image. Here we have also tested the performance of the models using different configurations as shown in Table 2. In all cases one has 794 visible units, corresponding to the $28^2 = 784$ pixels in each image, plus 10 additional units for the one-hot-encoded labels. Regarding the standard RBM, several models with 100, 200, 500 and 900 hidden units have been tested, shown in the third column of Table 2. . As a reference, $N_h = 500$ has often been employed in the literature [65]. Each model contains a different total number of training parameters, reported in the second column of Table 2. As in the 4 × 4 BS case, we have compared the performance of these RBMs to the results obtained with different PIW and RAPID models with similar number of TP. A systematic analysis as the one shown in Fig. 4 resulted in different combinations of number of patterns and hidden units reported in the fourth column of Table 2. For all these experiments we have done 20 independent trainings using CD20.

Due to its large size, it is not possible to calculate the exact KL of the networks when learning this problem. Therefore, we have chosen to report the accuracy when used as a classifier, corresponding to a standard supervised scenario for RBMs [66]. Results for the best average validation accuracies are reported in the last three columns of Table 2. Two numbers are reported in each case, the left ones corresponding to the result produced directly by the model, while the right one is the result of a logistic regression performed over the values of the hidden units, as in Ref. [43].

As can be seen, the PIW models perform better in all configurations. Furthermore, in some cases the improvements is remarkable, particularly on the results obtained from the bare models. Additionally, PIW yields similar results regardless of the number of TP, something that does not happen with the other two. In this way, one can conclude that the PIW architecture is more robust. Note also that PIW requires a lower number of TP than the standard RBM and RAPID to achieve similar or better results.

The average accuracy along learning of the validation set with the corresponding error bars for the bare PIW and RBM models with $4 \cdot 10^5$ TP are shown in Fig. 6. As it can be seen, the standard RBM improves

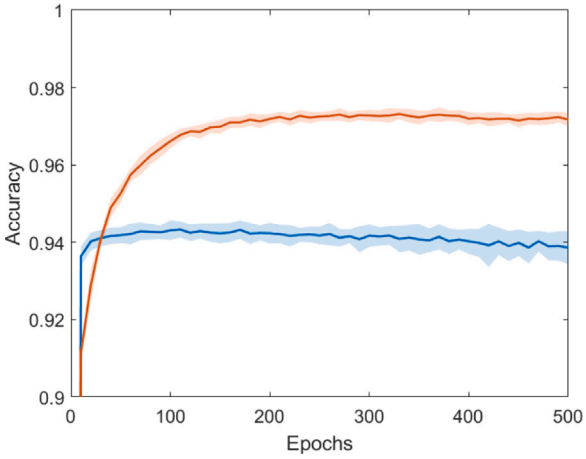


Fig. 6. Accuracy of PIW (red) and the standard RBM (blue) models in the validation set for the MNIST problem.

faster at a very early stage, but its best prediction is worse than the one achieved by PIW. From there on, both models start to slightly degrade, while PIW seems to be more stable.

Once trained, one can also check the performance of the PIW network as a generative model. In order to do that we use corrupted versions of test set images, where half the pixels are replaced by uniformly distributed ±1 values. We then apply 10 Gibbs steps to these vectors, keeping the non-corrupted part of the image and the associated label. Results for this experiment are shown in Fig. 7, where one can see that, overall, the reconstruction performance of the trained model is almost perfect. This confirms that the model has been able to capture the underlying features required for conditional generation.

6.3. Connect-4

We have also tested PIW, RAPID and the standard RBM in the Connect-4 dataset, which contains 16 000 training and 47 557 validation images of 126 binarized pixels with 3 different labels [62]. These images correspond to legal intermediate situations in the Connect-4 game with 7 × 6 positions board, while the three classes indicate the final result for player one (win, loss, draw). As in the MNIST case, this dataset is too large to compute the exact KL, so we once again resort to the evaluation of the accuracy instead.

As in the previous cases, we have performed an exhaustive search of model parameters, including the number of hidden units, learning rate and number of patterns in the PIW case, for fixed total number of TP. As before, four representative choices of a standard RBM have been used, containing 50, 100, 190 and 350 hidden units. In order to get a similar number of TP in the PIW and RAPID networks, we have selected the number of patterns and hidden units as shown in the table.

As in the MNIST case, Table 2 display the results of the best average validation accuracy of each model after 20 independent runs. Once again, the PIW models produce more stable predictions when changing the number of TP, while the standard RBM results show a slightly larger variation. In this case, the larger RBM models yield somewhat better results than PIW, although the improvement is not really relevant. In contrast, for the smaller models PIW performs slightly better. In all cases, though, the predictions are very similar and slightly better than RAPID.

Table 2
Accuracies obtained with the different RBM models for the problems tested. Columns two to four refer to the number of training parameters, hidden units and patterns. The last three columns report the results. Numbers on the left and right correspond to the accuracies both from bare models and the logistic regressions, respectively.

Dataset	TP	N_h RBM	K/N_h PIW & RAPID	RBM CD20	PIW CD20	RAPID CD20(ξ)
MNIST	$8 \cdot 10^4$	100	75/265	0.256/0.863	0.958/0.955	0.114/0.815
MNIST	$1.6 \cdot 10^5$	200	145/300	0.682/0.886	0.969/0.961	0.111/0.848
MNIST	$4 \cdot 10^5$	500	240/900	0.943/0.939	0.976/0.967	0.528/0.909
MNIST	$7 \cdot 10^5$	900	190/3000	0.964/0.960	0.971/0.969	0.558/0.932
Connect4	$6.6 \cdot 10^3$	50	28/100	0.710/0.704	0.743/0.710	0.670/0.703
Connect4	$1.3 \cdot 10^4$	100	25/350	0.744/0.729	0.765/0.780	0.725/0.759
Connect4	$2.5 \cdot 10^4$	190	50/350	0.770/0.768	0.767/0.783	0.685/0.745
Connect4	$4.4 \cdot 10^4$	350	90/350	0.784/0.790	0.767/0.783	0.713/0.765
OCR-Letters	$9.4 \cdot 10^3$	60	43/60	0.741/0.774	0.763/0.807	0.620/0.748
OCR-Letters	$3.9 \cdot 10^4$	250	85/300	0.849/0.861	0.862/0.874	0.511/0.858
OCR-Letters	$9.5 \cdot 10^4$	610	170/400	0.873/0.881	0.874/0.884	0.467/0.840

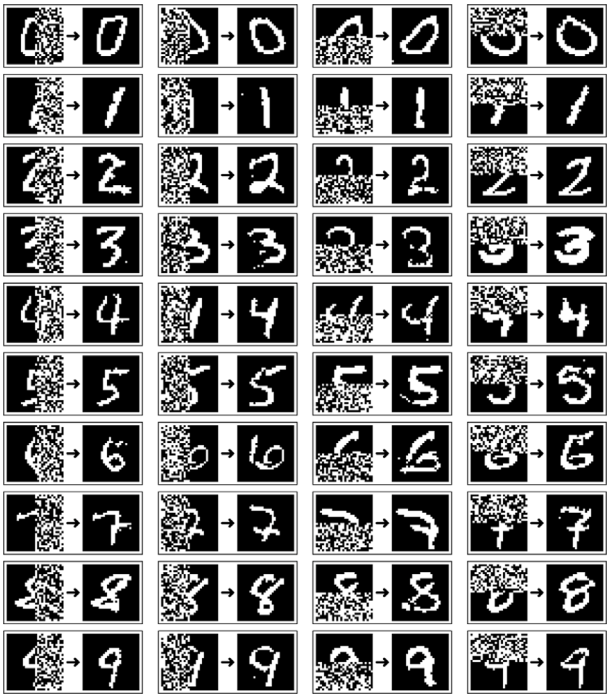


Fig. 7. Reconstruction of half-corrupted images from the MNIST dataset using the trained PIW model.

6.4. OCR-Letters

Finally, we compare the performance of the standard RBM, RAPID and the PIW models in the OCR-Letters dataset [63], composed by 32 152 training and 10 000 validation images of 128 binarized pixels with 26 different labels, leading to models with 154 visible units. As in the previous cases and due to its large size, we evaluate the quality of the models using the accuracy metric. Three standard RBMs with 60, 250 and 610 hidden units were tested. As before, a search with similar number of TP has yielded best results for PIW and RAPID with the parameters reported in Table 2. Once again, we have carried out 20 independent training runs in each case.

As seen in the last columns of the table, a similar behavior to the one found in the previous cases is observed here, with the PIW models producing more stable accuracies under a change of the number of TP, and overall better performance than the standard RBM and RAPID models.

7. Summary and conclusions

To summarize, in this work we analyze some of the main properties of the original RAPID model of Ref. [43] to identify potential improvements. We track the evolution of the Kullback–Leibler (KL) divergence along learning to find that, even in simple models where quantities can be derived analytically, the final values obtained are not as competitive as those produced by a standard RBM. Based on that we have relaxed some of the approximations of the RAPID model that significantly constrain its performance, while preserving the interesting idea of building network weights from patterns. We call this model PIW.

We have tested the PIW network on both small and large problems. We have found that, for small datasets where the KL can be directly computed, the final values obtained after learning are similar or slightly better than the values obtained training a standard RBM. In larger classification problems, where the KL cannot be exactly evaluated, we have observed a similar trend in accuracy, which is, on average, slightly better than that provided by a standard RBM. In all cases, PIW achieves better results than RAPID. Despite the differences in performance of PIW with respect to the standard RBM not being too significant, the main advantages of PIW are its requirement for a much lower number of TP, and the stability of its predictions under reasonable changes to network parameters.

CRediT authorship contribution statement

J. Garí: Writing – review & editing, Validation, Software, Methodology, Investigation, Conceptualization. **E. Romero:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Conceptualization. **F. Mazzanti:** Writing – review & editing, Writing – original draft, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

F. Mazzanti acknowledges financial support from the project PID 2020-113565GB-C21 funded by MCIN/AEI/10.13039/501100011033, and from Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya, co-funded by the European Union Regional Development Fund within the ERDF Operational Program of Catalunya (project QuantumCat, ref. 001-P-001644). E. Romero: This paper is part of project PID2022-143299OB-I00, financed by MCIN/AEI/10.13030/501100011033/FEDER, UE.

References

[1] P. Smolensky, Chapter 6: Information processing in dynamical systems: Foundations of harmony theory, in: D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition (Vol. 1), MIT Press, 1986, pp. 194–281.

[2] G.E. Hinton, T.J. Sejnowski, Learning and relearning in Boltzmann machines, in: D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition (Vol. 1), MIT Press, 1986, pp. 282–317.

[3] A. Fischer, C. Igel, Training restricted Boltzmann machines: An introduction, Pattern Recognit. 47 (1) (2014) 25–39, http://dx.doi.org/10.1016/j.patcog.2013.05.025.

[4] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, University of Toronto, 2012.

[5] K. Cho, T. Raiko, A. Ilin, Gaussian-Bernoulli deep Boltzmann machine, in: Proceedings of the International Joint Conference on Neural Networks, 2013, pp. 1–7, http://dx.doi.org/10.1109/IJCNN.2013.6706831.

[6] J. Zhang, H. Wang, J. Chu, S. Huang, T. Li, Q. Zhao, Improved Gaussian–Bernoulli restricted Boltzmann machine for learning discriminative representations, Knowl.-Based Syst. 185 (2019) 104911, http://dx.doi.org/10.1016/j.knosys.2019.104911, URL https://www.sciencedirect.com/science/article/pii/S0950705119303661.

[7] R. Liao, S. Kornblith, M. Ren, D.J. Fleet, G. Hinton, Gaussian-Bernoulli RBMs without tears, 2022, arXiv:2210.10318.

[8] J. Chu, J. Liu, H. Wang, H. Meng, Z. Gong, T. Li, Micro-supervised disturbance learning: A perspective of representation probability distribution, IEEE Trans. Pattern Anal. Mach. Intell. 45 (6) (2023) 7542–7558, http://dx.doi.org/10.1109/TPAMI.2022.3225461.

[9] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted Boltzmann Machines for Collaborative Filtering, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 791–798.

[10] Q. Fu, Y. Zhuang, J. Gu, Y. Zhu, X. Guo, Agreeing to disagree: Choosing among eight topic-modeling methods, Big Data Res. 23 (2021) 100173.

[11] Mixed-variate restricted boltzmann machines, J. Mach. Learn. Res. 20 (2011) 213–229.

[12] S. Ogawa, H. Mori, A Gaussian-Gaussian-restricted-Boltzmann-machine-based deep neural network technique for photovoltaic system generation forecasting, IFAC-PapersOnLine 52 (4) (2019) 87–92, http://dx.doi.org/10.1016/j.ifacol.2019.08.160, IFAC Workshop on Control of Smart Grid and Renewable Energy Systems CSGRES 2019, URL https://www.sciencedirect.com/science/article/pii/S2405896319304963.

[13] A. Decelle, C. Furtlehner, Gaussian-spherical restricted Boltzmann machines, J. Phys. A 53 (18) (2020) 184002, http://dx.doi.org/10.1088/1751-8121/ab79f3.

[14] G.E. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.

[15] W. Deng, H. Liu, J. Xu, H. Zhao, Y. Song, An improved quantum-inspired differential evolution algorithm for deep belief network, IEEE Trans. Instrum. Meas. 69 (10) (2020) 7319–7327.

[16] I. Sohn, Deep belief network based intrusion detection techniques: A survey, Expert Syst. Appl. 167 (2021) 114170.

[17] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[18] J. Sleeman, J. Dorband, M. Halem, A hybrid quantum enabled RBM advantage: convolutional autoencoders for quantum image compression and generative learning, in: Quantum Information Science, Sensing, and Computation XII, Vol. 11391, SPIE, 2020, pp. 23–38.

[19] J. Chu, H. Wang, J. Liu, Z. Gong, T. Li, Multi-local collaborative AutoEncoder, Knowl.-Based Syst. 239 (2022) 107844, http://dx.doi.org/10.1016/j.knosys.2021.107844, URL https://www.sciencedirect.com/science/article/pii/S0950705121010327.

[20] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 609–616, http://dx.doi.org/10.1145/1553374.1553453.

[21] D.A. Puente, I.M. Eremin, Convolutional restricted Boltzmann machine aided Monte Carlo: An application to ising and kitaev models, Phys. Rev. B 102 (19) (2020) 195148.

[22] I. Sutskever, G.E. Hinton, G.W. Taylor, The recurrent temporal restricted Boltzmann machine, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), in: Advances in Neural Information Processing Systems, vol. 21, Curran Associates, Inc., 2008, URL https://proceedings.neurips.cc/paper_files/paper/2008/file/9ad6aaed513b73148b7d49f70afcfb32-Paper.pdf.

[23] L. Xia, J. Lv, C. Xie, J. Yin, A conditional classification recurrent RBM for improved series mid-term forecasting, Appl. Intell. 51 (11) (2021) 8334–8348.

[24] K. Raza, N.K. Singh, A tour of unsupervised deep learning for medical image analysis, Curr. Med. Imag. 17 (9) (2021) 1059–1077.

[25] S. Suganthi, M.U.A. Ayoobkhan, N. Bacanin, K. Venkatachalam, H. Štěpán, T. Pavel, et al., Deep learning model for deep fake face recognition and detection, PeerJ Comput. Sci. 8 (2022) e881.

[26] R. Kirubahari, S.M.J. Amali, An improved restricted Boltzmann machine using Bayesian optimization for recommender systems, Evol. Syst. (2023) 1–13.

[27] A.-R. Mohamed, G.E. Dahl, G. Hinton, Acoustic Modeling using Deep Belief Networks, IEEE Trans. Audio Speech Lang. Process. 20 (1) (2012) 14–22.

[28] G. Carleo, M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science 355 (6325) (2017) 602–606.

[29] R.G. Melko, G. Carleo, J. Carrasquilla, J.I. Cirac, Restricted Boltzmann machines in quantum physics, Nat. Phys. 15 (9) (2019) 887–892.

[30] M. Neugebauer, L. Fischer, A. Jäger, S. Czischek, S. Jochim, M. Weidemüller, M. Gärttner, Neural-network quantum state tomography in a two-qubit experiment, Phys. Rev. A 102 (4) (2020) 042604.

[31] R. Savitha, A. Ambikapathi, K. Rajaraman, Online RBM: Growing restricted Boltzmann machine on the fly for unsupervised representation, Appl. Soft Comput. 92 (2020) 106278.

[32] A. Decelle, C. Furtlehner, Restricted Boltzmann machine: Recent advances and mean-field theory, Chin. Phys. B 30 (4) (2021) 040202.

[33] N. Béreux, A. Decelle, C. Furtlehner, B. Seoane, Learning a restricted Boltzmann machine using biased Monte Carlo sampling, SciPost Phys. 14 (3) (2023) 032.

[34] T. Liu, S. Chen, K. Li, S. Gan, C.J. Harris, Adaptive multioutput gradient RBF tracker for nonlinear and nonstationary regression, IEEE Trans. Cybern. (2023).

[35] T. Liu, Z. Tian, S. Chen, K. Wang, C.J. Harris, Deep cascade gradient RBF networks with output-relevant feature extraction and adaptation for nonlinear and nonstationary processes, IEEE Trans. Cybern. (2022).

[36] T. Liu, S. Chen, S. Liang, S. Gan, C.J. Harris, Fast adaptive gradient RBF networks for online learning of nonstationary time series, IEEE Trans. Signal Process. 68 (2020) 2015–2030.

[37] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (2002) 1771–1800.

[38] A. Fischer, C. Igel, Empirical analysis of the divergence of gibbs sampling based learning algorithms for restricted Boltzmann machines, in: K. Diamantaras, W. Duch, L.S. Iliadis (Eds.), Artificial Neural Networks – ICANN 2010, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 208–217.

[39] Y. Bengio, O. Delalleau, Justifying and generalizing contrastive divergence, Neural Comput. 21 (6) (2009) 1601–1621.

[40] T. Tieleman, G.E. Hinton, Using fast weights to improve persistent contrastive divergence, in: 26th International Conference on Machine Learning, 2009, pp. 1033–1040.

[41] K. Cho, T. Raiko, A. Ilin, Parallel tempering is efficient for learning restricted Boltzmann machines, in: IEEE International Joint Conference on Neural Networks, 2010, pp. 1–8.

[42] E. Romero, F. Mazzanti, J. Delgado, D. Buchaca, Weighted contrastive divergence, Neural Netw. 114 (2019) 147–156.

[43] A. Pozas-Kerstjens, G. Muñoz-Gil, E. Piñol, M.A. García-March, A. Acín, M. Lewenstein, P.R. Grzybowski, Efficient training of energy-based models via spin-glass control, Mach. Learn.: Sci. Technol. 2 (2) (2021) 1601–1621, http://dx.doi.org/10.1088/2632-2153/abe807.

[44] A. Barra, A. Bernacchia, E. Santucci, P. Contucci, On the equivalence of hopfield networks and Boltzmann machines, Neural Netw. 34 (2012) 1–9.

[45] E. Agliari, A. Barra, A. De Antoni, A. Galluzzi, Parallel retrieval of correlated patterns: From hopfield networks to Boltzmann machines, 38, 2013, pp. 52–63.

[46] C. Marullo, E. Agliari, Boltzmann machines as generalized hopfield networks: a review of recent results and outlooks, Entropy 23 (1) (2020) 34.

[47] E. Agliari, G. Sebastiani, Learning and retrieval operational modes for three-layer restricted Boltzmann machines, 185, (2), 2021.

[48] M. Smart, A. Zilman, On the mapping between hopfield networks and restricted Boltzmann machines, in: ICLR 2021 - 9th International Conference on Learning Representations, 2021.

[49] E. Agliari, F. Alemanno, A. Barra, G. De Marzo, The emergence of a concept in shallow neural networks, Neural Netw. 148 (2022) 232–253.

[50] E. Agliari, A. Barra, C. Longo, D. Tantari, Neural networks retrieving boolean patterns in a sea of Gaussian ones, J. Stat. Phys. 168 (2017) 1085–1104.

[51] J. Tubiana, R. Monasson, Emergence of compositional representations in restricted Boltzmann machines, Phys. Rev. Lett. 118 (13) (2017).

[52] D.H. Ackley, G.E. Hinton, T.J. Sejnowski, A learning algorithm for Boltzmann machines, Cogn. Sci. 9 (1985) 147–169.

[53] Y. Bengio, Learning Deep Architectures for AI, Found. Trends Mach. Learning 2 (1) (2009) 1–127.
[54] K.H. Fischer, J.A. Hertz, Spin glasses, (no. 1) Cambridge University Press, 1993.
[55] D.J. Amit, Modeling Brain Function: The World of Attractor Neural Networks, Cambridge University Press, 1989, http://dx.doi.org/10.1017/CBO9780511623257.
[56] D. Sherrington, S. Kirkpatrick, Solvable model of a spin-glass, Phys. Rev. Lett. 35 (1975) 1792–1796, http://dx.doi.org/10.1103/PhysRevLett.35.1792.
[57] J.A. Hertz, A. Krogh, R.G. Palmer, Introduction to the theory of neural computation., The advanced book program, vol. 1, Addison-Wesley, 1991.
[58] D.O. Hebb, The organization of behavior: A neuropsychological theory, Wiley, New York, 1949.
[59] J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. USA 79 (1982) 2554–2558.
[60] D.J. MacKay, Information theory, inference, and learning algorithms, vol. 7, Cambridge University Press, 2003.
[61] Y. LeCun, C. Cortes, MNIST handwritten digit database, 2010, URL http://yann.lecun.com/exdb/mnist/.
[62] Connect-4 data set, URL https://archive.ics.uci.edu/ml/datasets/Connect-4.
[63] OCR-Letters data set, URL http://ai.stanford.edu/~btaskar/ocr.
[64] A. Fischer, C. Igel, Empirical Analysis of the Divergence of Gibbs Sampling Based Learning Algorithms for Restricted Boltzmann Machines, in: International Conference on Artificial Neural Networks, Vol. 3, ICANN, 2010, pp. 208–217.
[65] R. Salakhutdinov, I. Murray, On the Quantitative Analysis of Deep Belief Networks, in: International Conference on Machine Learning, 2008, pp. 872–879.
[66] H. Larochelle, Y. Bengio, Classification using discriminative restricted Boltzmann machines, in: Proceedings of the 25th International Conference on Machine Learning, 2008, pp. 536–543.

Jon Garí received the B.S. degree in Physics (Theoretical Physics mention) from the Universitat de Barcelona in 2019, and the M.S. degree in Engineering Physics from the Universitat Politècnica de Catalunya, Barcelona, in 2021, where he is currently pursuing the Ph.D. degree in Computational and Applied Physics. His current research interests include energy-based models, machine learning and deep learning.

Enrique Romero received the B.Sc. degree in Mathematics in 1989 from the Universitat Autònoma de Barcelona, in Spain. In 1994, he received the B.Sc. degree in Computer Science from the Universitat Politècnica de Catalunya (UPC). In 1996, he joined the Department of Computer Science, UPC, where he works as an associate professor. In 2004, he received the Ph.D. degree in Computer Science from the UPC. His research interests include Pattern Recognition, Support Vector Machines, Neural Networks and Deep Learning.

Ferran Mazzanti Castrillejo received the B.Sc. and Ph.D. degrees in physics from the Universitat de Barcelona, Barcelona, Spain, in 1991 and 1997, respectively. In 1992, he joined the Electronics Department, at "Enginyeria i Arquitectura La Salle", Barcelona, as an Associate Professor. In 2006, he moved to the Physics Department of the 'Universitat Politècnica de Catalunya', in Barcelona. His current research interests include quantum many body problems at zero and finite temperature, computational simulation of quantum systems, Boltzmann machine neural networks, and deep learning.