INSTITUT POLYTECHNIQUE DE PARIS - ENSTA PARIS

Restricted Boltzmann Machine

Florian MOREL
Tao GUINOT

2024-2025

1 Introduction

Restricted Boltzmann Machines (RBMs) are generative stochastic artificial neural networks capable of learning a probability distribution over a set of inputs. Originally introduced by Smolensky in 1986, RBMs gained prominence through their application in various machine learning tasks, such as dimensionality reduction, classification, collaborative filtering, feature learning, and generative modeling.

The RBM architecture consists of two layers : a visible layer that interacts with the input data and a hidden layer that captures dependencies between input features. The interaction between these two layers is characterized by symmetric connections, with no intra-layer communication. The training process typically leverages contrastive divergence, a popular and efficient approximation method, to optimize the model's parameters.

In this report, we explore the potential of RBMs in modeling and reconstructing image datasets. By utilizing different datasets, we assess the performance of RBMs in terms of their ability to generate accurate representations of input data. Our study provides insights into the behavior of RBMs under various hyperparameter configurations and their capacity to generalize across diverse datasets.

2 Dataset

For this study, several datasets are used to train the RBM and generate images from it. Let's describe them. It is important to note that we binarise all the images. For those being greyscaled (FashionMNIST) we therefore set to 1 pixels having a value between $127 - 255$ and to 0 pixels having a value between $0 - 126$.

2.1 Binary Alpha Digits

The Binary Alpha Digits dataset consists in 36×39 images of size 20×16 covering numbers from $0 - 9$ and letters from the latin alphabet. (fig. 1)

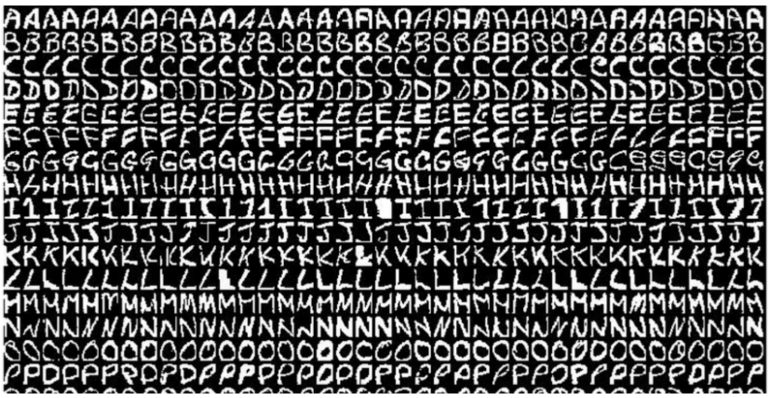

FIGURE 1 – Binary Alpha Digits dataset

2.2 MNIST

The MNIST dataset is composed of handwritten digits including 60,000 traning examples and 10,000 test examples. (fig. 2)

FIGURE 2 – MNIST Dataset

2.3 FashionMNIST

Fashion-MNIST is a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. (fig. 3)

FIGURE 3 – FashionMNIST Dataset

Here is an example of a binarized image from this dataset, representing a heel shoe. (fig. 4)

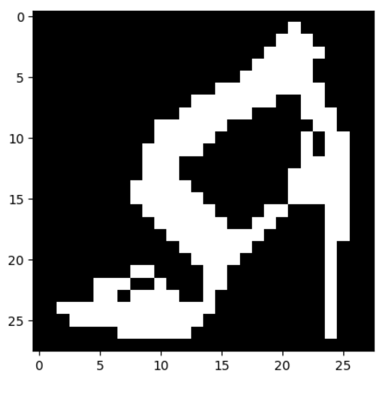

FIGURE 4 – Binarized image from FashionMNIST

3 Model

RBM use the principle of Gibs sampling to approximate the distribution of images on which we train the model thanks to a latent space. We note V the variable corresponding to an input image, and H a latent variable.

The goal is to set a starting value for v and alternatively compute $p(h|v)$ and $p(v|h)$. Indeed, we can express $p(h|v)$ and $p(v|h)$ as :

$$p(h|v) = sigmoid(b_j + \sum_i w_{ij} v_i)$$

$$p(v|h) = sigmoid(a_i + \sum_j w_{ij} h_j)$$

With $a_i, b_j, w_i j$ corresponding to weights describing an Energy term :

$$E_\theta(v,h) := -(\sum_{i=1}^{p} a_i v_i + \sum_{i,j} w_{ij} v_i h_j + \sum_{j=1}^{q} b_j h_j)$$

p being the dimension of the input image and q the dimension of the latent space.

We will not describe precisely the RBM algorithm.

The model is initialized with weights following a normal distribution with 0 as mean and 10^{-1} as variance. Biases are set to 0.

For parameters, we use *nn.Parameter* to automatically add variable to the list of model's parameters. This also tells Pytorch to include the corresponding tensor in the computation graph and compute gradients for it during backpropagation.

4 Training and Experiments

4.1 Training

The training is quite straightforward as we go through epochs to alternate the steps described earlier (computing $p(h|v)$ and $p(v|h)$). It is important to notice that we shuffle the dataset before iterating through batches to simulate a random draw.

To update parameters during training, we normalize the gradient to the batch size. We use the following reconstruction error :

$$error = \frac{sum(X - X_{reconstructed}^2)}{n \times p}$$

With n the size of the image, p the size of the latent space.

4.2 Experiments

We first led experimentations on hyperparameters tuning. As a reminder, the interesting hyperparameters are the following :

- q : dimension of the latent space
- batch size
- number of character the learn (only numbers, only letters, both, etc.)

4.2.1 Dimension of the latent space

To some extent, the deeper a neural network is, the better the model's performance. The objective of this section is to study the performance of the RBM, in terms of reconstruction error, as a function of the number of hidden states, represented by the variable q. We have therefore trained five different RBMs with a varying number of hidden units, and we obtain the results shown in Figure 5 after the generation phase.

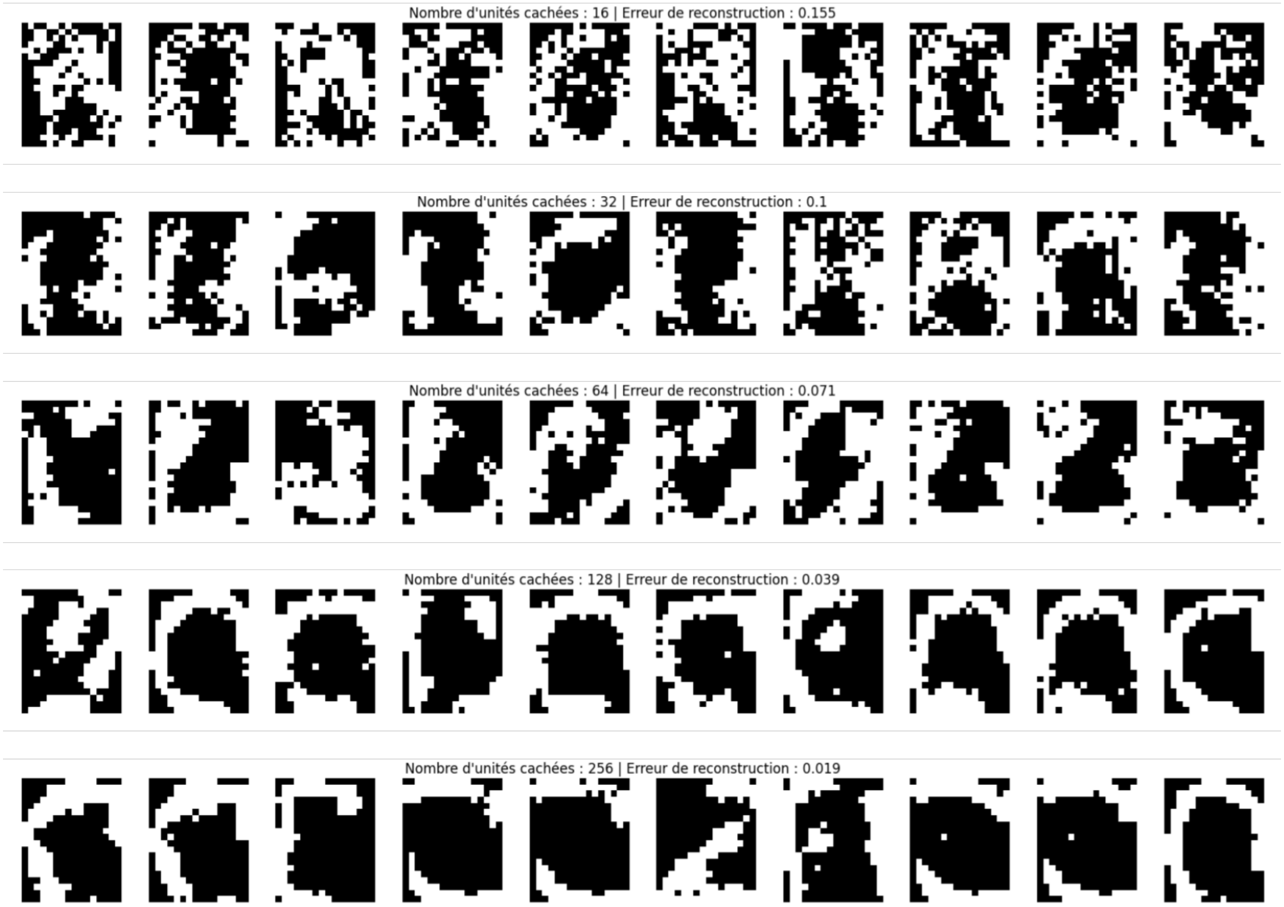

FIGURE 5 – Influence of the number of hidden states

Similarly to what can be observed in the study of deep neural networks, we notice that the dimension of the latent space, i.e., the number of hidden states, plays a major role in the quality of generation. The larger it is, the lower the reconstruction error. Qualitatively, we observe sharper images for a large q, where different elements found in the training dataset, such as `0` or `2` or `C`, can be distinguished.

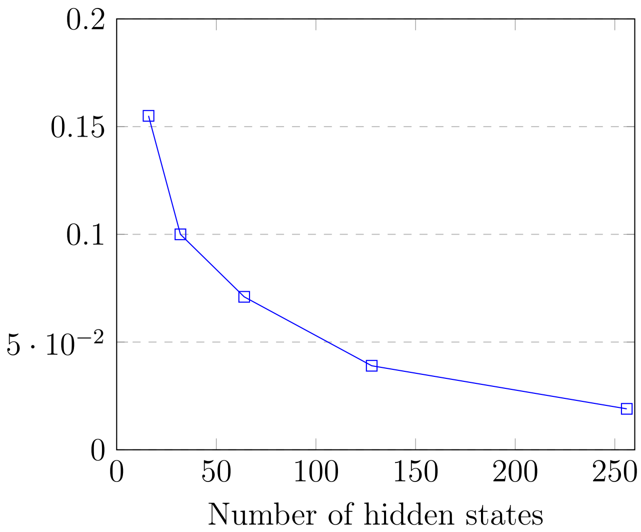

FIGURE 6 – Reconstruction error regarding the number of hidden states

By plotting the evolution of the reconstruction error as a function of the number of hidden states considered (Figure 6), we observe that the curve converges around an error of 0.019 for approximately 256 hidden states. Due to our limited computing resources, we were unable to test an RBM with 512 hidden states, which could potentially be the optimally tuned value for the hyperparameter q.

4.2.2 Influence of the batch size

The second hyperparameter we studied for optimization is the batch size. Similar to the previous study, we trained multiple RBMs with different batch sizes and obtained the qualitative and quantitative results shown in Figure 7.

"Qualitatively, it is easy to detect a very poor generation when the batch size is only 1 image (first line). This can be explained by overfitting, which causes the RBM to reproduce exactly the images it was trained on, and then overlap them, forming the white masses that can be observed. In contrast, a very large batch size (100 images or more) prevents the model from properly learning the subtleties of the images to be generated. These images are not sharp, and apart from a generic 'O', the model is not able to generate images that exist in the initial space.

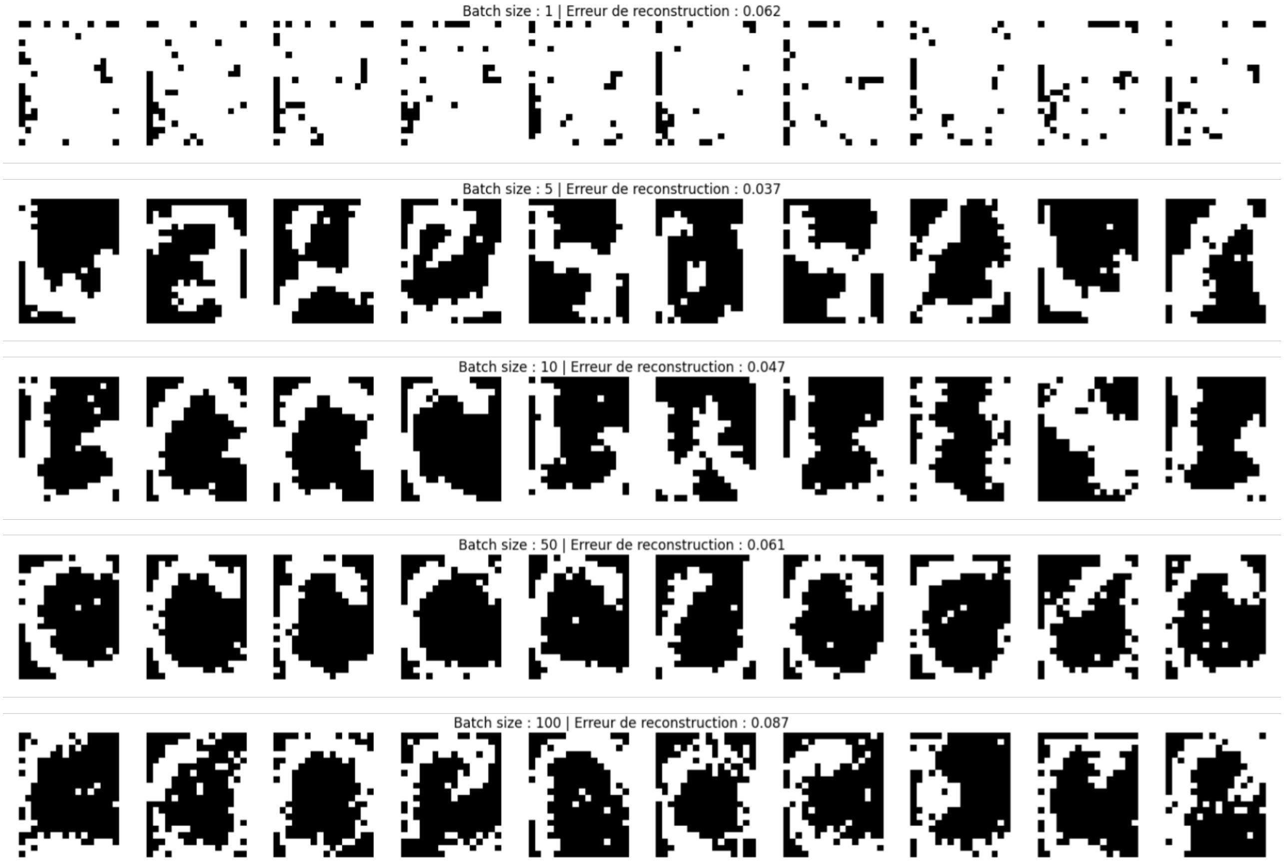

FIGURE 7 – Influence of batch size on the quality of generation

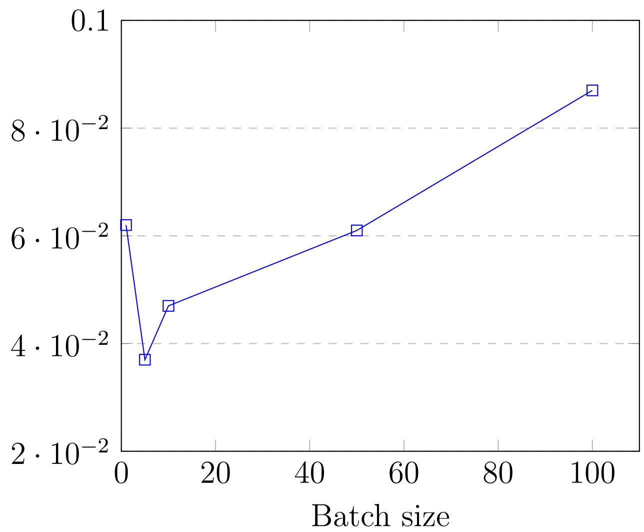

FIGURE 8 – Reconstruction error regarding the batch size used during the training step

The graph (Figure 8), which displays the reconstruction error as a function of batch size, clearly shows that the ideal batch size is around 5. In Figure 7, it is at this value that we can most clearly distinguish images that are close to the training dataset, like characters `W`, `3` or `4`.

4.2.3 Number of character to learn

Another interesting study to conduct is the behavior of the RBM based on the number of characters it is given to integrate, in other words, the size of the distribution of images to be generated. We therefore trained 26 different RBMs, such that RBM number i was trained to generate the first i letters of the alphabet. The quantitative results are shown in Figure 9.

FIGURE 9 – Influence of batch size on the quality of generation

The qualitative results show that when the model has fewer characters at its disposal, it can more easily focus on them and generate them more cleanly. In the first three lines, where A, AB, and ABC are considered respectively, these characters are easily distinguishable. In the last two lines, where A to M and A to Z are respectively considered, the generated characters are more difficult to distinguish.

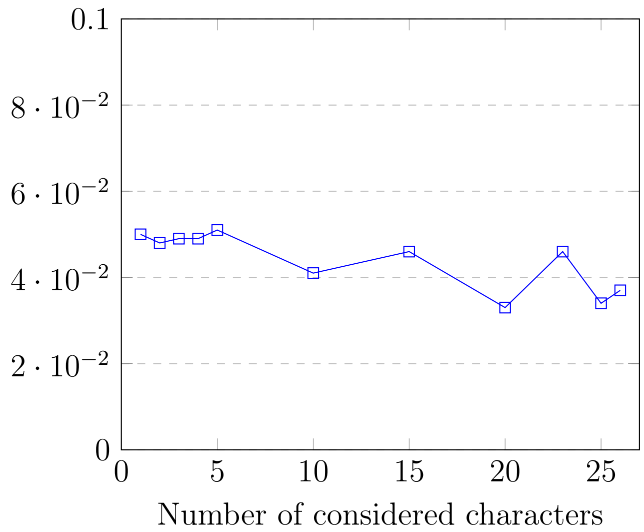

FIGURE 10 – Reconstruction error regarding the number of considered characters to learn

The quantitative study shown in Figure 10 reveals that the number of characters to learn does not significantly affect the reconstruction error. The slight decreasing slope can be explained by an increase in the dataset size when an additional character is added. With more examples, the model learns to generalize better.

The gap observed between the quantitative and qualitative results may lead to a reconsideration of the relevance of the metric used, the reconstruction error.

4.3 Other datasets and Other models

In order to estimate the actual performance of the RBM, it is interesting to apply it to different datasets and compare it to other models capable of performing the same task.

4.3.1 Datasets

MNIST Another well-known dataset similar to binary alpha digits is the MNIST dataset. We therefore trained an RBM to generate images from this dataset, and the results are shown in Figure 11. To make a fair comparison with the previous dataset and to reduce computation time, we only kept 1404 random images from MNIST. Despite a reconstruction error similar to what we found previously, the generated images are not very meaningful, except for the 0, which is also predominant in the binary alpha digits dataset. This can be explained by the need for a study of the hyperparameters, especially for this dataset, to obtain better results. We did not take the time to conduct this study.

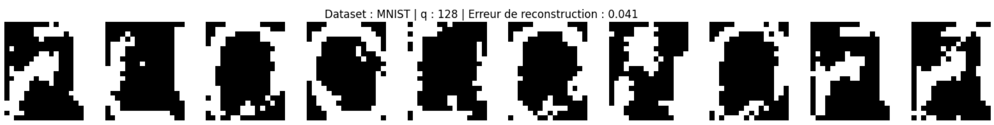

FIGURE 11 – RBM on the MNIST dataset

Fashion-MNIST To move away from images representing numbers or digits and to make the task more complex, we tested the RBM on the Fashion-MNIST dataset. The results are shown in 10 generated images in Figure 12. During generation, the results show an omnipresence of the shirt, or, if we are less optimistic, a white square. These mixed performances can be explained by the lack of parameter optimization or by the increasing complexity of the task. Unlike what we saw previously, the elements to be generated can have larger areas of white pixels. The pattern does not depend on the passage of a pencil.

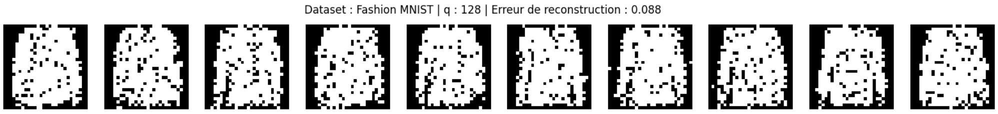

FIGURE 12 – RBM on the Fashion-MNIST dataset

4.3.2 GAN model

The goal of this section is to compare the RBM with more recent models, such as the GAN. We couldn't find a way to quickly train a GAN on the binary alpha digits dataset. However, it is possible to get an idea of its performance based on a GitHub repository, which has already trained it for us on the MNIST dataset. The results are showned in Figure 13.

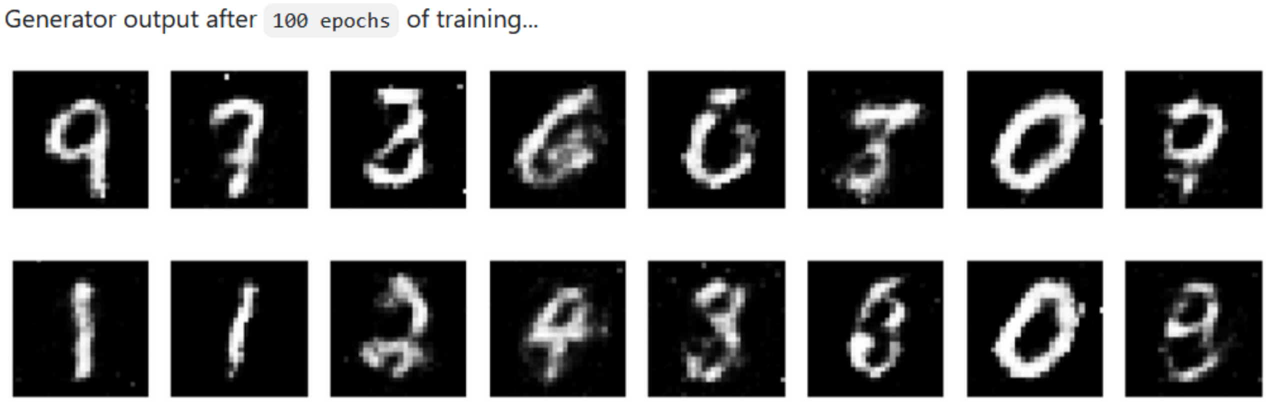

FIGURE 13 – Example of generated images with a GAN on the MNIST dataset

Although the study is conducted on non-binary images, the results show that they are much cleaner than those generated by our RBM, and the characters are more easily recognizable. This explains the rise of models such as VQ-VAE, GANs, and diffusion models in recent years, to the detriment of more classical models like RBMs.