CAN NEURAL-NETWORKS LEARN A GAUGE SYMMETRY?

A. Decelle, L.A. Fernández , V. Martín-Mayor, **B. Seoane***

* beaseobar@gmail.com

Introduction

During the past decades, the field of Machine Learning has been very successful in designing mathematical models and learning algorithms to learn automatically how to modelize data in order to obtain latent representation leading to incredible performance both in classification tasks and in generating credible new data from a data-set. Such powerful tools have potentially a broad application in many other fields, and in particular in Statistical Physics. In this work, we explore the possibility of using neural networks to predict the thermodynamic properties of the spin glass Edwards-Anderson model, just by looking at the interaction network. If such a machine existed, it should be able to detect a complex non-local symmetry that couples the physical properties of large groups of disorder realizations (samples): a gauge symmetry. In this work, we tackle the problem of designing a neural network able to distinguish if two samples are, or not, related by a gauge transformation.

Definitions

The 2D Edwards-Anderson model is defined on a square Euclidean lattice by the following Hamiltonian

$$\mathcal{H} = -\sum_{\langle i,j\rangle} J_{ij} s_i s_j \qquad (1)$$

The spins $s_i = \pm 1$ are dynamical variables and lie on the N lattice sites, and the couplings $J_{ij} = \pm 1$ are random but quenched variables and are placed on the $2N$ edges of the lattice. The actual distribution of J_{ij} in each realization defines a *sample*.

The Hamiltonian (1) is invariant under the following *gauge transform*

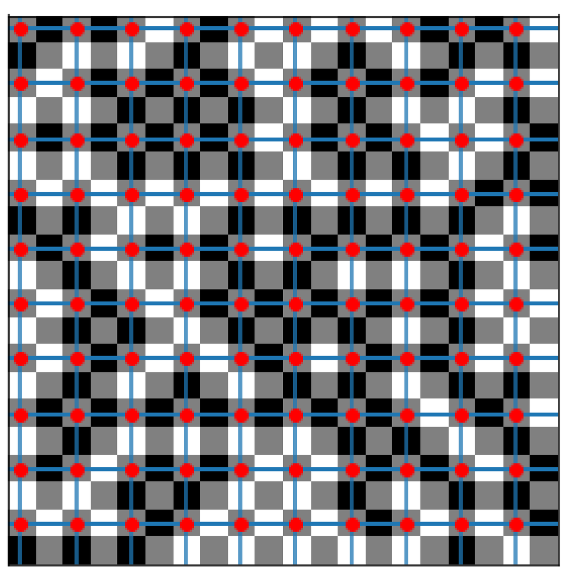
$$s_i' = \epsilon_i s_i \qquad J_{ij}' = \epsilon_i \epsilon_j J_{ij} \qquad \forall i,j \in N$$

for any possible choice of $\{\epsilon_i = \pm 1\}$, which means that all the samples related by this symmetry (**same gauge orbit**) have identical thermodynamic properties.

This symmetry can be formulated in a different way: if two samples are gauge symmetric, **the product of J_{ij} along any closed loop remain unchanged !** This makes this symmetry particularly difficult to identify since it is not enough to look at local properties (such as small "plaquette") !

Deep Neural Network and Data-set

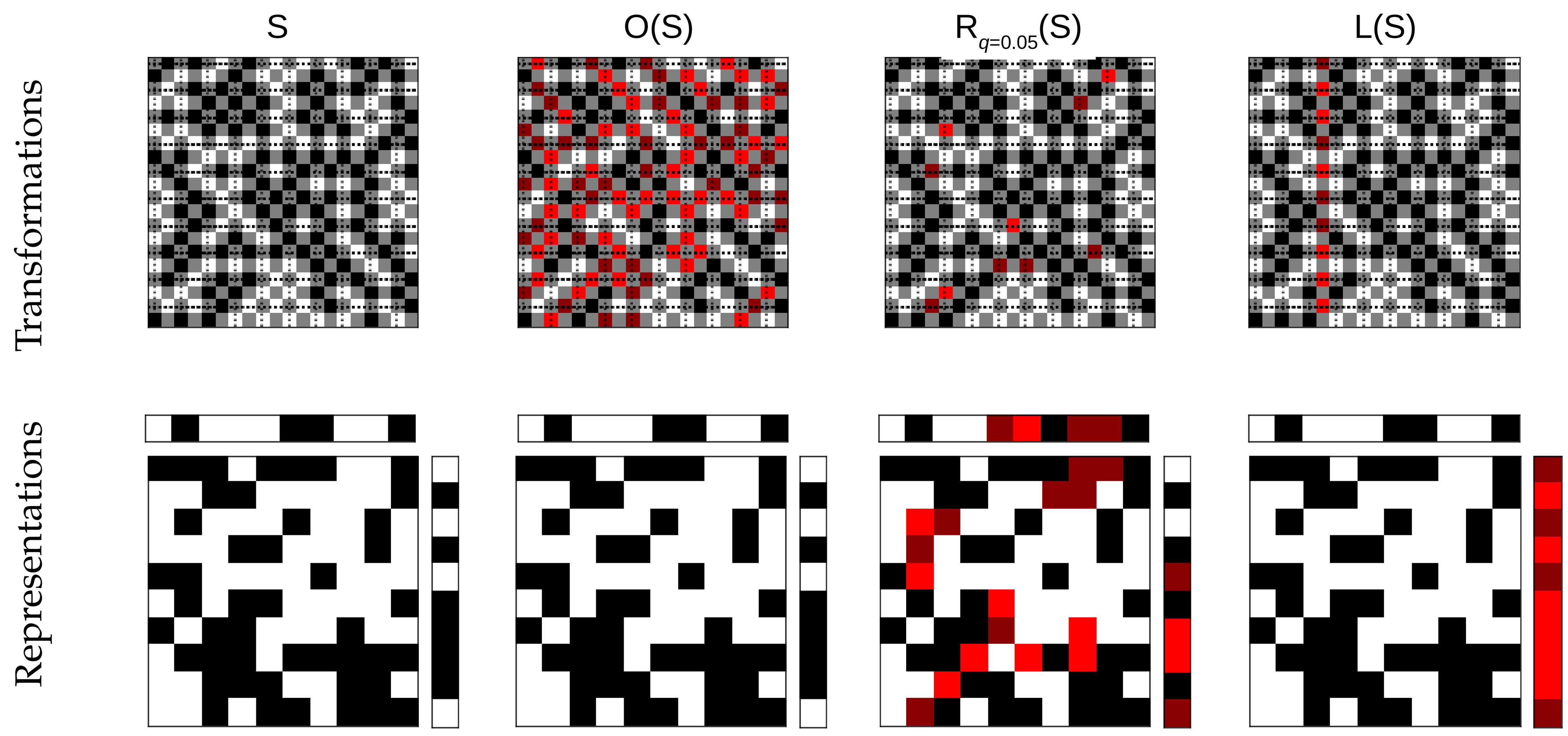
Deep neural networks (DNNs) are particularly adapted to deal with problems on regular lattices. In particular, convolutional layers are often used as a features detector on images. In our work, we used the Keras library in order to test the ability of NNs to recognize the Gauge Symmetry. In order to use all these tools, we first need to transform our set of interactions in images to be scanned by the NN.



The simplest solution is illustrated in the figure, where the original spins of the lattice are shown as red dots as a guide for the eye. In our new image, the pixels in between two spins are black (white) if the interaction is ferromagnetic (antiferromagnetic). The rest of useless pixels are fixed to zero (grey) and are equal for all the samples.

Transformations and training set

We want to feed the DNN with two samples (A) and (B), and get as output if they are or not on the same gauge orbit. With this goal, the DNN is trained with a set of N_{train} couples. Beyond the simplicity of this test, **most of the DNN tested fail to infer the gauge symmetry (they only find trivial classifications), unless a "smart" architecture is introduced in the DNN !**

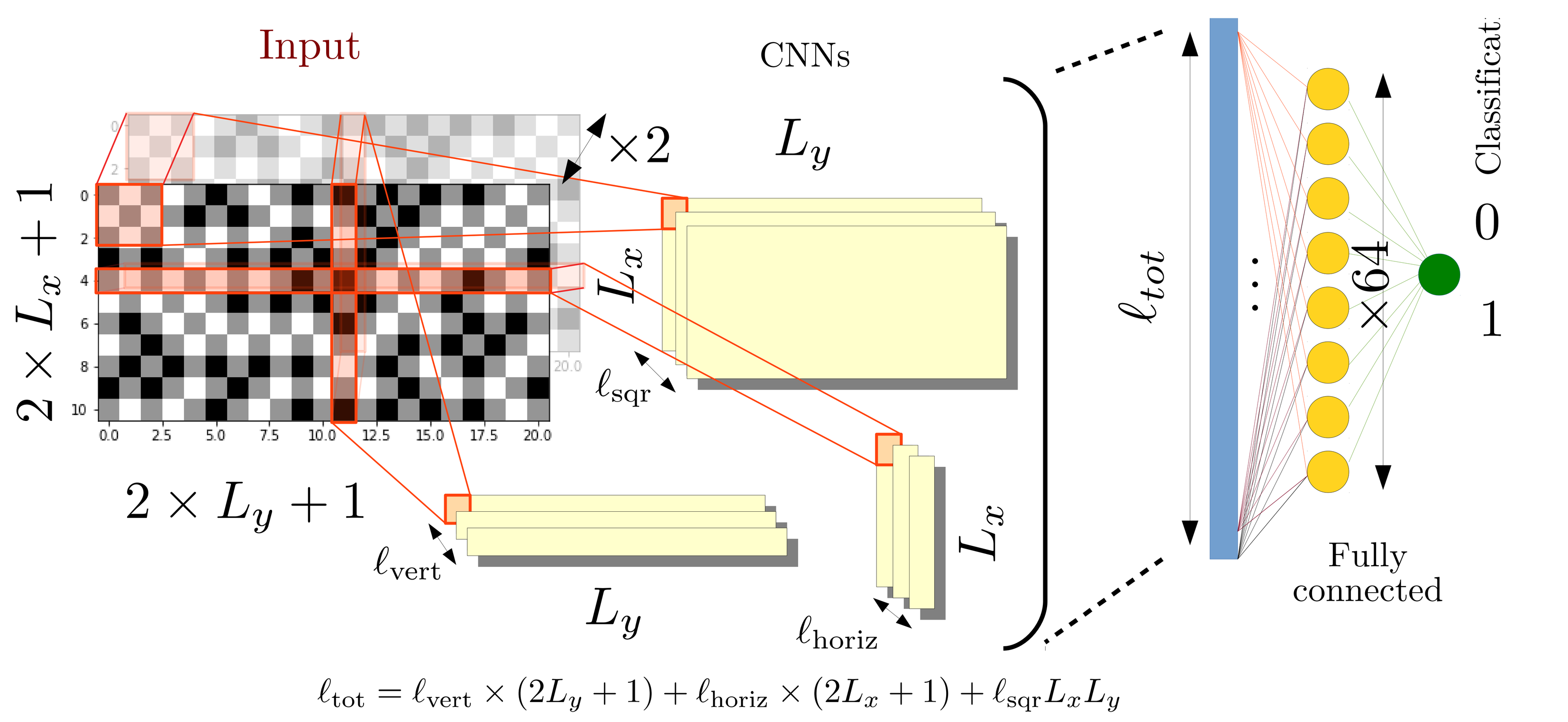


Case 1: S-O(S) and S-S′
This case is easy: the DNN only needs to compare the value of a reduced number of plaquettes. For this, a DNN with a square (3×3) convolutional can succeed doing it. But this is not enough to detect more elaborated symmetry breaking.

Case 2: S-O(S) and S-R$_q$(S)
We can force the previous machine to check all the plaquettes by comparing random gauge transformations of two samples that differ only by a few links. Now, the learning is harder, but the DNN succeeds to classify this. Yet, the gauge symmetry is non-local.

Case 3: Line transformation
The transformation L(S) leaves all the plaquettes unaltered but it is not a gauge transformation. When dealing with these samples, the simple convolutional layer is not able anymore to properly classify all the samples. We need to check long loops.

Final architecture and results



In order to force the DNN to learn the gauge symmetry in a 2D Edwards-Anderson model it is necessary to use a proper training set to enforce the learning of useful latent features. A non-gauge transformation implies breaking at least one loop in the system, and any closed loop can be built from the sum of small local loops, the plaquettes, and/or large straight loops, e.g. the vertical/horizontal lines (under periodic boundary conditions). For this reason, we use the following architecture for the NN: each sample is scanned by a square kernel, a full line (horizontal) kernel and a vertical one. The image is fed (in parallel) to three different convolutional networks. These CNN serves as feature detectors before a fully-connected layer performs the classification.