Comparison of Three Models for Predicting Thermodynamic Quantities of a Spin Glass

Ernesto Pedrazzini
Artificial Intelligence for Science and Technology
Università degli Studi di Milano Bicocca
Email:e.pedrazzini4@campus.unimib.it

***Abstract*—The aim of this paper is to analyze and compare the performance of a Feed Forward Neural Network, a Custom Connected Neural Network and a Convolutional Neural Network for predicting the evolution of mean energy and Edwards-Anderson order parameter in function of the temperature in a dataset generated with the simulation annealing method.**

1. Introduction

Ising spin glasses are known to be NP-hard problems for classical computers: the frustration caused by random interactions between spins makes the energy landscape irregular, with many local minima.
For this reason, the use of neural networks might be beneficial for predicting the evolution of thermodynamic quantities in a spin glass, thanks to their ability to generalize from large datasets and provide results in a short computational time.

2. Dataset description

I decided to consider a spin glass with non periodic boundary conditions on a square lattice $N = 6 \times 6$.
Connections values among the neighbors of spin are taken so that $J_i \in \{+1, -1\}$, so there exist $2^{(6-1)\times 6\times 2}$ possible configurations of $\{J_1, J_2, \ldots, J_{60}\}$: from

$\sum_{k=1}^{60} J_k = -60$ to $\sum_{k=1}^{60} J_k = 60$.

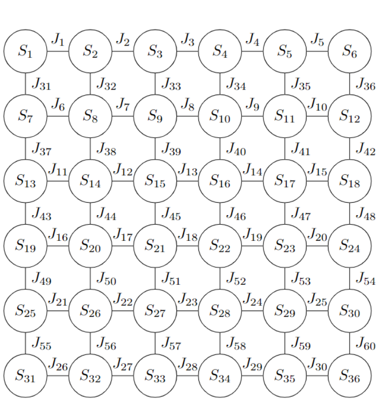

Figure 1. Spin glass 6×6 configuration.

In the dataset 800 configurations are generated uniformly with respect to all possible values of the sum of all interactions.
For each configuration, 41 temperatures from 4 to 0.1 in step 0.1 are generated, and for each temperature step, the initial spin configuration is allowed to thermalize for 10000 Monte Carlo steps, then in the next $n = 100000$ the energy value of the spin glass is calculated to determine the mean energy at that given temperature. Moreover, after thermalization, spins configurations are saved and, in the end, the Edwards-Anderson order parameter is computed according to the following formulas:

$$H = -\sum_{\langle i,j\rangle} J_k S_i S_j$$

$$\langle E\rangle = \frac{1}{n}\sum_{j=1}^{n} H_j$$

$$\langle q_{EA}\rangle = \frac{1}{N}\sum_i \langle \sigma_i\rangle^2$$

The temperature decreases so to reproduce the simulation annealing process and reach a more precise minimum for the energy.

After this process, two dataset are created: one will be passed to the CNN and it's composed by PyTorch tensors of shape $(6, 6, 4)$ in which any component corresponds to the site of a spin in the 6×6 grid, and is related to 4 numbers: the four J_{ij} connections with the four nearest neighbors of the site. To this configurations are added the information on the temperature T, the mean energy $\langle E\rangle$ and $\langle q_{EA}\rangle$ computed for that temperature for a total of 32800 configurations (800×41) with thermodynamic quantities associated.
Then, the other dataset is a simple reorganization of the first one: the configurations are represented by vectors of length 61 where J_i are ordered as in $Figure 1$, and the thermodynamic quantities are added as before. This last dataset is more suitable to be passed to the other two Neural Networks.

3. First model: Feed Forward Neural Newtork

The Feed Forward Neural Network is organized as in $Figure 2$.

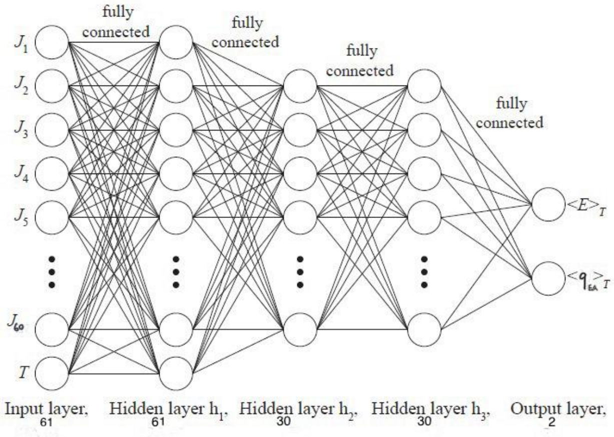

Figure 2. Feed Forward Neural Network structure.

The input is formed by the vector shaped configuration of the spin glass, with the addition of the information on the temperature. Then is passed to the first hidden layer with the same number of neurons, after that, the next two hidden layers present half of the neurons and the output layer is of course formed by two neurons: the labels $\langle E\rangle$ and $\langle q_{EA}\rangle$ to be learnt.
Between all the layers the activation function is Tanh, and is important to notice that the total number of trainable parameters is 6634, since will be used to compare it to the others two models.
After 30 epochs, the validation loss doesn't present any sign of over fitting as we can see in the next plot.

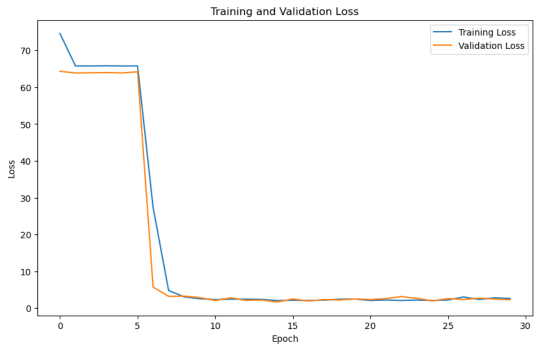

Figure 3. Training loss vs validation loss.

Performance are evaluated by the root mean squared error (RMSE) of the mean energy and the Edwards-Anderson order parameter and the results are:
RMSE $\langle E\rangle$: 2.0683
RMSE $\langle q_E A\rangle$: 0.1876.

4. Second model: Custom Connected Neural Network

The second model starts with an architecture similar to the first one, but I cut some of the weights between the first hidden layer and the second hidden layer, not randomly, but in a way that provides information about the spatial distribution of connections in the spin grid.
The architecture is schematically represented in $Figure 4$.

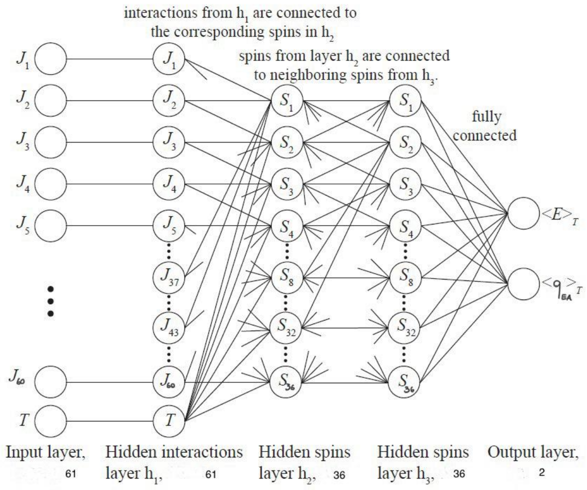
interactions from h$_1$ are connected to the corresponding spins in h$_2$
spins from layer h$_2$ are connected to neighboring spins from h$_3$.
fully connected
J_1
J_2
J_3
J_4
J_5
J_{37}
J_{43}
J_{60}
T
S_1
S_2
S_3
S_4
S_8
S_{32}
S_{36}
$<E>_T$
$<q_{EA}>_T$
Input layer, 61
Hidden interactions layer h$_1$, 61
Hidden spins layer h$_2$, 36
Hidden spins layer h$_3$, 36
Output layer, 2

Figure 4. Custom Connected Neural Network structure.

In this case, any of the neurons in the first hidden layer represents a connection between two spins, and the second hidden layer represent the spins. So, for example, the neuron in the tenth position, which represents the connection between S_{11} and S_{12} (see $Figure 1$), have non zero weights only with the 11^{th} and 12^{th} neuron of the second hidden layer. In the code it can be done using a mask: a sparse matrix with non zero elements only in positions that represent the weights connecting the neurons of interest. After the definition of the model, weights are initialized with the mask, then the backward step compute all the gradients, and the optimization step is modified so to update only weights corresponding to the positions of ones in the mask.
The total number of trainable parameters is much lower than in the Feed Forward Neural Network: 1720, but, after 30 epoch, the performance results are better:
RMSE $\langle E\rangle$: 1.6834
RMSE $\langle q_{EA}\rangle$: 0.1532.
This is due to the fact that even if I cut some of the weights, the network benefits from having information on the spatial distribution of the spin glass.
Of course one could improve the speed of the training exploiting the sparseness of the weight matrix, without computing the gradients of frozen weights, but I live it as a future improvement.

5. Third model: Convolutional Neural Network

In my opinion a natural solution to the problem that leverages the spatial distribution of the spin grid could be the usage of a convolutional neural network. In this case the input passed to the network is the $(6, 6, 4)$ shaped tensor produced in the first dataset. I attempted to design an architecture with a comparable number of trainable parameters to the Feed Forward Neural Network, and investigated whether this approach could still improve performances. The network is composed by two convolutional layers and two fully connected layers.
The kernel size is always 3×3, the layer depth increases from 4 to 32.
The first fully connected layer receives a flatten output from the convolutional layers composed by 129 neurons and at this stage I add one neuron containing the temperature value of the input configuration. The output layer is composed of course by 2 neurons.
The activation function used is a leaky ReLu and the structure obtained presents 7874 parameters to be tuned. After 30 epoch the RMSE is better than the other two models:
RMSE $\langle E\rangle$: 1.2172
RMSE $\langle q_{EA}\rangle$: 0.1281.
After the training of each model I generated a brand new configuration using simulation annealing, and compared the evolution of the energy and Edwards-Anderson order parameter with the predictions of the models to verify if the shape was correctly learned. Here I reported an example in $Figure 5$.

6. Conclusions

In this study, I have explored and compared three different neural network architectures for predicting thermodynamic quantities in a spin glass system. The results indicate that the CNN outperforms both the FFNN and CCNN in terms of prediction accuracy, as demonstrated by the lower root mean square error (RMSE) values for both the mean energy and the Edwards-Anderson order parameter.
The success of the CNN can be attributed to its ability to effectively capture the spatial correlations in the spin glass through the use of convolutional layers.
It's interesting how in the CCNN the number of parameters can be drastically reduced, however is not always possible to find a way to provide information to a network pruning some of the weights.

References

[1] Petr Andriushchenko, Dmitrii Kapitan, Vitalii Kapitan, *A New Look at the Spin Glass Problem from a Deep Learning Perspective*, 2022, https://doi.org/10.3390/e24050697

[2] Andrew Lucas, *Ising formulations of many NP problems*, 2014, https://doi.org/10.3389/fphy.2014.00005

[3] V.S. Dotsenko, *Physics of the spin-glass state*, 1993

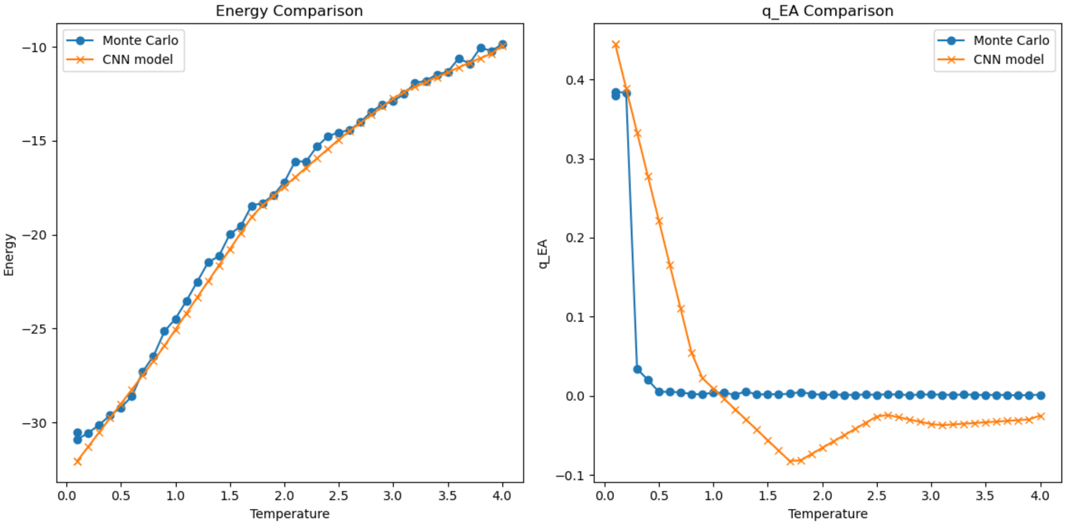

Figure 5. Monte Carlo method vs CNN predictions.