Mini-project: Residual entropy of a frustrated magnet

Submitted to Markus Schmitt and Jonas Rigo by Moksh Bhateja

March 08, 2024

The goal of this project is to plot entropy as a function of temperature for a 2D antiferromagnetic triangular lattice with periodic boundary conditions. The approach involves utilizing auto-regressive neural networks (RNNs) to minimize free energy, enabling us to estimate the system's entropy.

1 Introduction

Magnetic systems are essential subjects of study in physics, offering insights into complex phenomena. The Ising model applied to ferromagnetic interactions on a 2D square lattice has been solved analytically will serves as a benchmark/reference to test the models/frameworks. Then we will investigate antiferromagnetic 2D triangular lattice (where spins preferentially align in opposite directions, introducing additional complexity) using these models/frameworks. To understand the behavior of such systems advanced models such as autoregressive neural networks (RNNs) are used.

1.1 Ising model

The Ising model is a simple, classical lattice model of a ferromagnet. In its simplest form, it is defined in terms of classical spins (s_j) taking on the values (± 1) on a square lattice. The Hamiltonian of the Ising model can be written as

$$H_J(s) = -J \sum_{<i,j>} s_i s_j$$

where J is dimensionless parameter. The sum is over all nearest-neighbor pairs on the lattice, with each pair counted once. The parameter J represents the coupling of each spin to its nearest neighbors. If $J > 0$ the coupling is ferromagnetic and in the limit $T \to 0$ i.e. all spins are aligned, and for $J < 0$ the coupling is antiferromagnetic. In this study we will assume $|J| = 1$.
At a given inverse temperature $\beta = \frac{1}{T}$, the probability of a spin configuration s is given by the Boltzmann distribution

$$p_\beta(s) = \frac{1}{Z} e^{-\beta H_J(s)}$$

Where the partition function, Z is given by

$$Z = \sum_s e^{-\beta H_J(s)}$$

and the statistical average of any observable X is given by

$$\langle X \rangle = \sum_s X p_\beta(s)$$

Central quantity of interest in statistical mechanics is the free energy,

$$F = -\frac{1}{\beta} \ln Z = E - \frac{1}{\beta} S$$

with the energy $E = \langle H(s) \rangle_{s \sim p_\beta}$ and the entropy $S = -\langle \ln p_\beta(s) \rangle_{s \sim p_\beta}$.

1.2 Autoregressive Neural Network and Markov Chains

Autoregressive neural networks (RNNs) play a pivotal role in capturing the conditional probability of a spin configuration given its predecessors. These networks are instrumental in approximating the Maxwell-Boltzmann probability distribution, especially when dealing with the unknown partition function. The implementation of Markov chains through RNNs ensures the generation of realistic lattice configurations, considering the historical states of the spins.

The autoregressive probability distribution for a given lattice configuration (s). is expressed as a product of conditional probabilities:

$$q_\theta(s) = q_\theta^{(1)}(s_1) q_\theta^{(2)}(s_2|s_1) q_\theta^{(3)}(s_3|s_1, s_2) \dots q_\theta^{(N)}(s_N|s_1, \dots, s_{N-1})$$

The chosen ansatz model $q_\theta(s)$ is constrained to be normalized:

$$\sum_s q_\theta(s) = 1$$

Model Architectures

Three different RNN model architectures were used:

Model A takes the lattice as input and outputs the probability $q_\theta(s)$. This model is independent of temperature. (fig. 1)

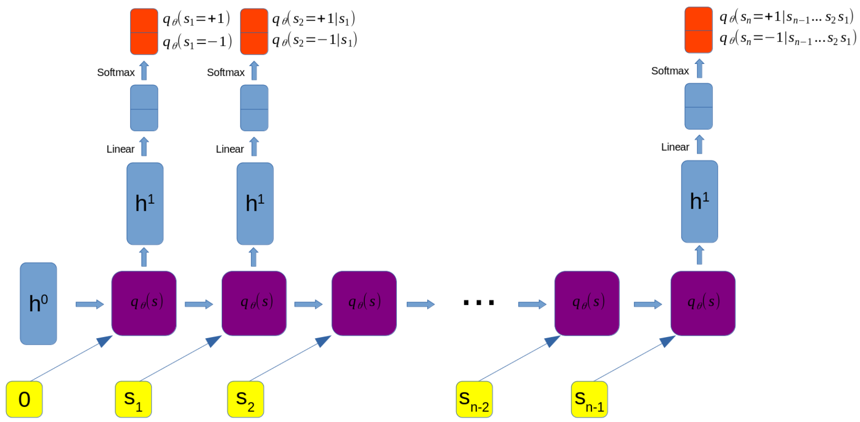


Figure 1: Model A architecture.

Model B also takes $\beta = \frac{1}{T}$ as an additional input. (fig. 2)

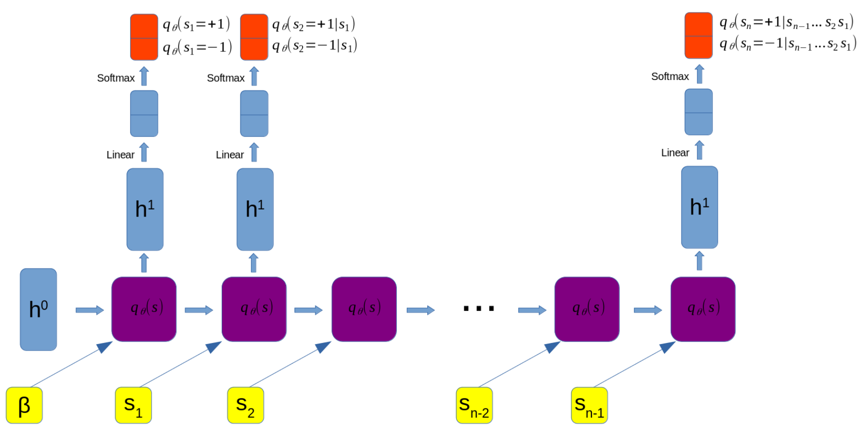


Figure 2: Model B architecture.

Model C is similar to Model B but embeds β differently in the architecture. (fig. 3)

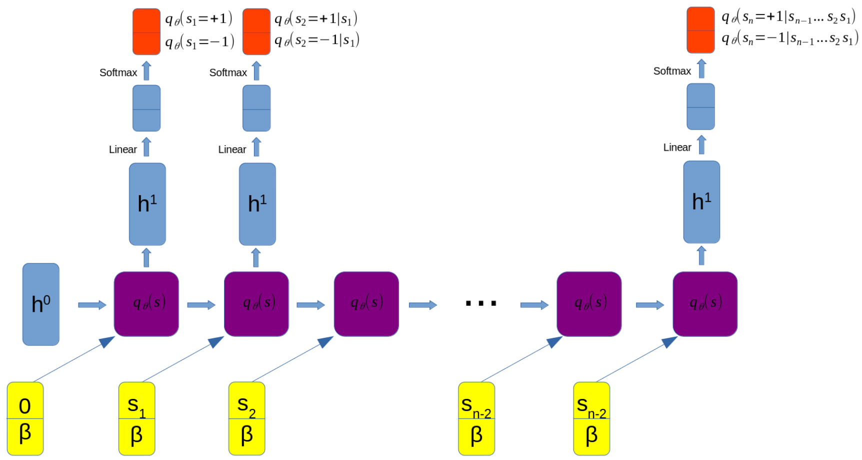


Figure 3: Model C architecture.

Models were trained with the following settings: input size=1, hidden size=70, num layers=2

1.3 Metropolis Algorithm

The Metropolis algorithm is a Monte Carlo method widely used for sampling from probability distributions. It was introduced by Metropolis et al. in 1953 [2] and has since become a fundamental tool in statistical physics, Bayesian statistics, and various other fields. This report provides a detailed overview of the Metropolis algorithm, its applications, and practical considerations.

Algorithm Description

The Metropolis algorithm is a Markov Chain Monte Carlo (MCMC) method designed to generate a sequence of samples from a target probability distribution. The basic steps of the algorithm are as follows:

Algorithm 1 Metropolis Algorithm

```
Initialize the system in a random state x
for t = 1 to T (number of iterations) do
    Propose a new state x' based on a proposal distribution
    Calculate the acceptance probability A(x, x')
    Generate a uniform random number u from [0, 1]
    if u ≤ A(x, x') then
        Accept the proposed state: x ← x'
    Record the current state x as a sample
```

Here, $A(\mathbf{x}, \mathbf{x}')$ is the acceptance probability, and its specific form depends on the target distribution and the proposal distribution.

The Ising model Metropolis algorithm is widely used to simulate the evolution of a magnetic system on a lattice and for studying magnetic phase transitions, critical phenomena, and thermal behavior of magnetic materials. The spins of the system are represented by variables that can take values ± 1. The algorithm to sample configurations from the Boltzmann distribution follows these steps:

Algorithm 2 Metropolis Algorithm for the Ising Model

```
Initialize the system with spins {s_i} on a lattice
for t = 1 to T (number of iterations) do
    Choose a random spin site i on the lattice
    Flip the spin at site i: s_i → −s_i
    Calculate the change in energy: ΔE = 2 s_i Σ_{neighbors j} s_j
    if ΔE ≤ 0 or exp(−ΔE/k_B T) > rand() then
        Accept the new spin configuration
    else
        Reject the new spin configuration: s_i → −s_i (undo the flip)
    Record the current spin configuration
```

Here, k_B is the Boltzmann constant and the neighbors of spin i are determined by the lattice structure. The size of the lattice affects the computational cost and the ability to capture macroscopic behavior. Larger lattices may be necessary for accurate results, but they also increase computational demands.

1.4 2D Ising model on rectangular lattice

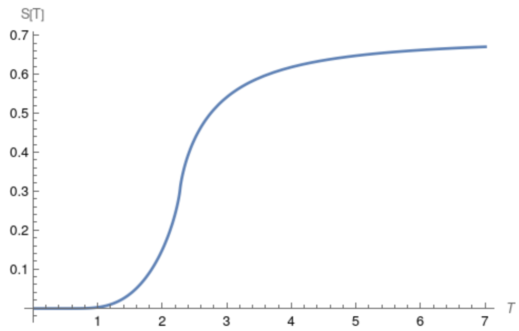
2D Ising model on rectangular lattice is well known and admits an exact solution [1] for the partition function $(Z(k))$ with:

$$W(k) = -\ln Z(k) = -\frac{1}{2}\ln(2\sinh(2k)) - \int_0^{\pi} \frac{dw}{2\pi} \,\mathrm{arcCosh}\left[\cosh(2k)\cosh(2K(k)) - \cos(w)\right]$$

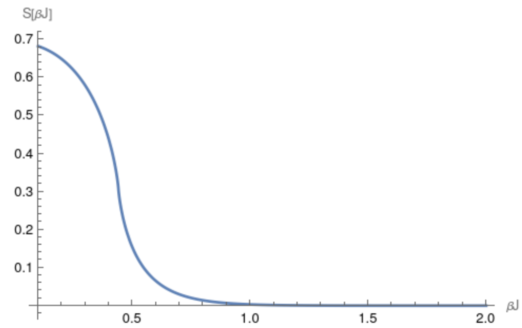
where $K(k) = \frac{1}{2}\ln\coth k$ and $k = J/k_B T$. The entropy $(\mathcal{S}_a)$ is then given by

$$\mathcal{S}_a = -\left(W(k) - k\frac{dW(k)}{dk}\right)$$

One can plot this equation and realize that $\mathcal{S}_a \to 0$ as $T \to 0$



(a) Entropy ($\mathcal{S}_a$) vs temperature (T)



(b) Entropy ($\mathcal{S}_a$) vs temperature (βJ)

2 Obejctive

Our objective of the RNN model is to minimize the free energy, F_θ, given by the expression:

$$F_\theta = \frac{1}{\beta} \sum_s q_\theta(s) \left(\beta H_J(s) + \ln q_\theta(s)\right)$$

Here, H_J denotes the Hamiltonian or energy of the configuration with nearest neighbor interactions.

If we are only dealing with lattices at equilibrium (i.e. only the lattices with a large and constant $q_\theta(s) \approx p_\beta(s)$ at a given temperature) then the free energy can be written as,

$$\beta F_\theta \approx \frac{1}{\text{Batch size}} \sum_s \left(\beta H_J(s) + \ln q_\theta(s)\right)$$

Where the summation is carried over a batch of configurations $S_q = \{s^{(1)}, \ldots, s^{(M)}\}$ where $s^{(i)} \sim q_\theta$. Once we have trained the model $q_\theta(s)$, the expectation of the Hamiltonian E is approximated as:

$$E \approx \langle H_J(s) \rangle_{s \sim q_\theta}$$

and the entropy S can be approximated (because $q_\theta(s) \approx p_\beta(s)$) as the average logarithm of the autoregressive probability distribution q_θ:

$$S \approx \langle \ln q_\theta(s) \rangle_{s \sim q_\theta}$$

To optimize the model two variants of loss function were minimized:

1.) "Mean Squared Error" loss:

$$L_1(\theta) = \left(\beta H_J(s) + \ln q_\theta(s)\right)^2$$

2.) By taking log of free energy

$$L_2(\theta) = \begin{cases} \ln q_\theta(s) + \ln\left(\beta H_J(s) + \ln q_\theta(s)\right), & \text{if } \left(\beta H_J(s) + \ln q_\theta(s)\right) \geq 0 \\ -[\ln q_\theta(s) + \ln\{-\left(\beta H_J(s) + \ln q_\theta(s)\right)\}], & \text{otherwise} \end{cases}$$

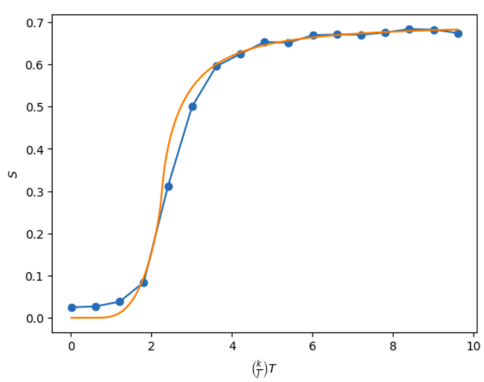
3 Approach and Results

3.1 Approach 1: Metropolis Algorithm for Data Generation

In the first approach, the Metropolis algorithm is employed to generate training data. This dataset comprises of thermally equilibrated lattice configurations and their corresponding energies. This involves running the Metropolis algorithm for a certain duration to achieve thermalization at a specific β. In this study the algorithm runs for 100001 iterations and only the last 10000 iterations are used as training set. For "Model A" a set of 17 different β values were chosen and for each value a different model was trained, since the Model A architecture is independent of β. Models A, B, and C are then trained using this data, with each model outputting the probability distribution q_θ for the given lattice configuration. These probabilities are crucial in minimizing the free energy through a delicate balance between energy and entropy. However, this method is memory-intensive due to storing the training data (lattice configurations). Entropy is normalized with N^2 where N^2 is the number of lattice sites.

Following results were obtained after 14 epochs with 17000 data length (64 batch size) and $L_2(\theta)$: (Orange line represents the theoretical plot)

3.1.1 2D Ising model for ferromagnetic rectangular lattice



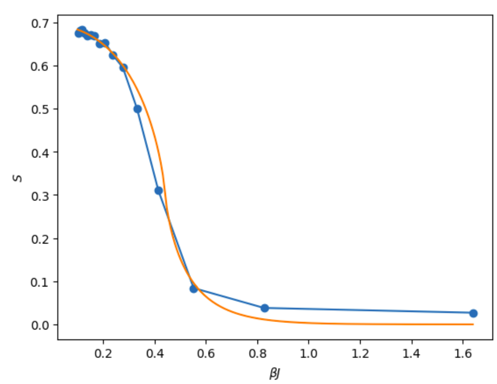


Figure 5: Entropy per site as function of T and βJ for Model A with 1st approach and N=6

Min entropy value 0.02475624 for Model A

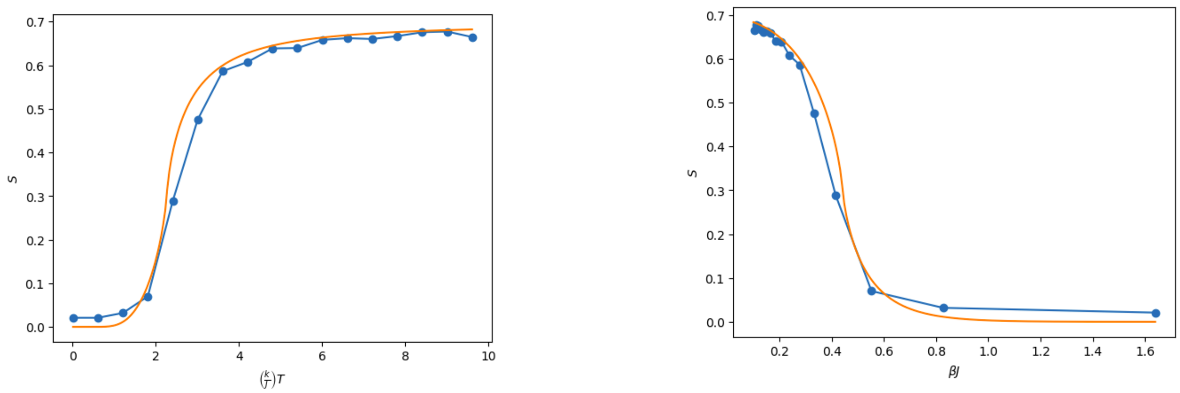


Figure 6: Entropy per site as function of T and βJ for Model B with 1st approach and N=6

Min entropy value 0.02092374 for Model B

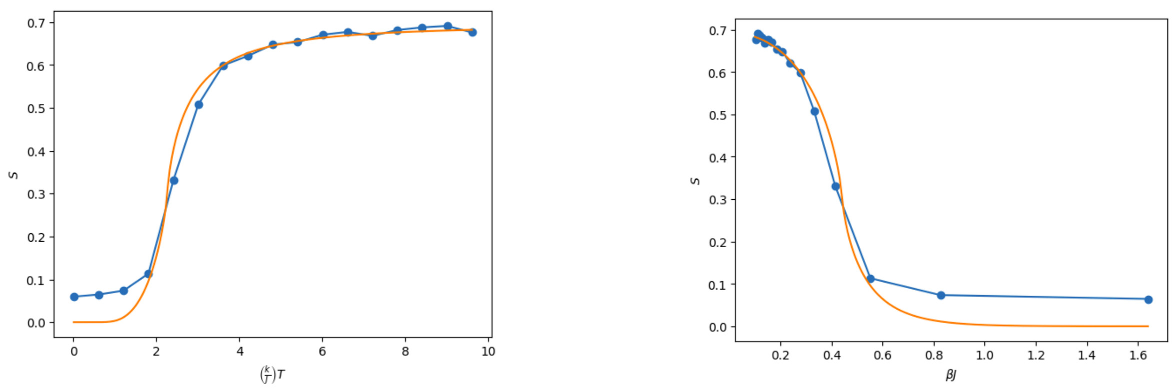


Figure 7: Entropy per site as function of T and βJ for Model C with 1st approach and N=6

Min entropy value 0.05957698 for Model C

3.1.2 2D Ising model for antiferromagnetic triangular lattice

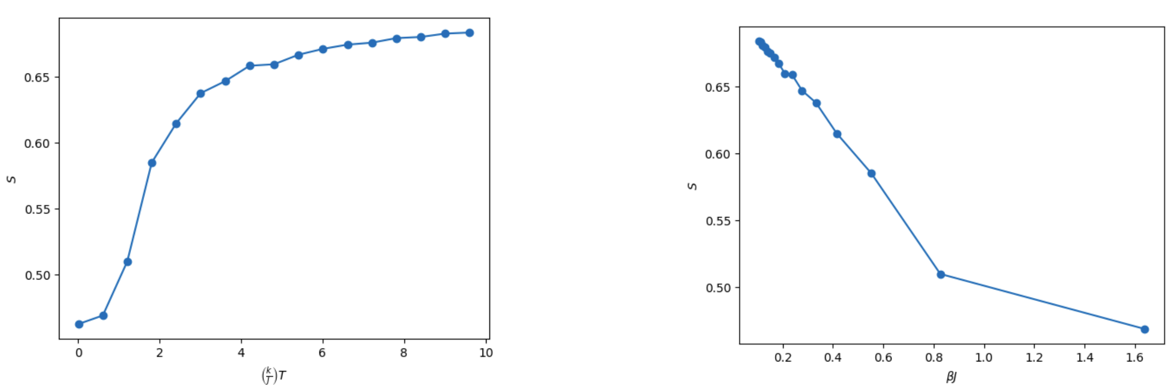


Figure 8: Entropy per site as function of T and βJ for Model A with 1st approach and N=6

Min entropy value 0.46214485 for Model A

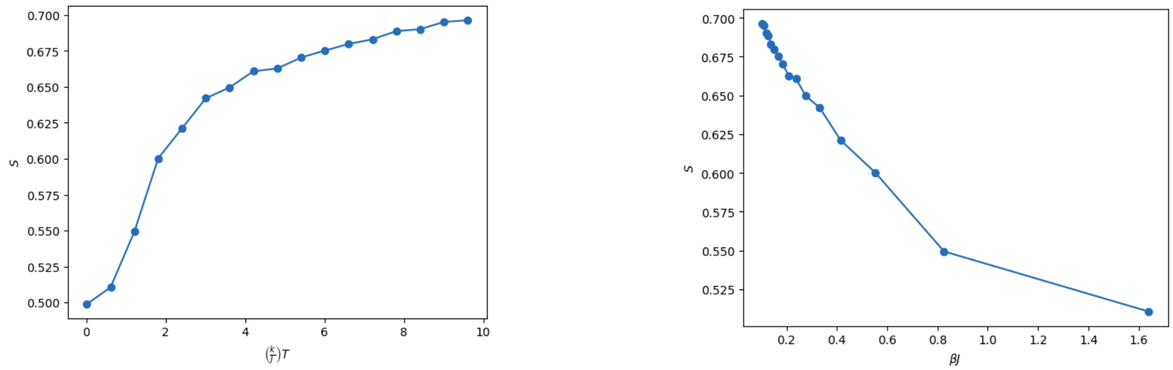


Figure 9: Entropy per site as function of T and βJ for Model B with 1st approach and N=6

Min entropy value 0.49887678 for Model B

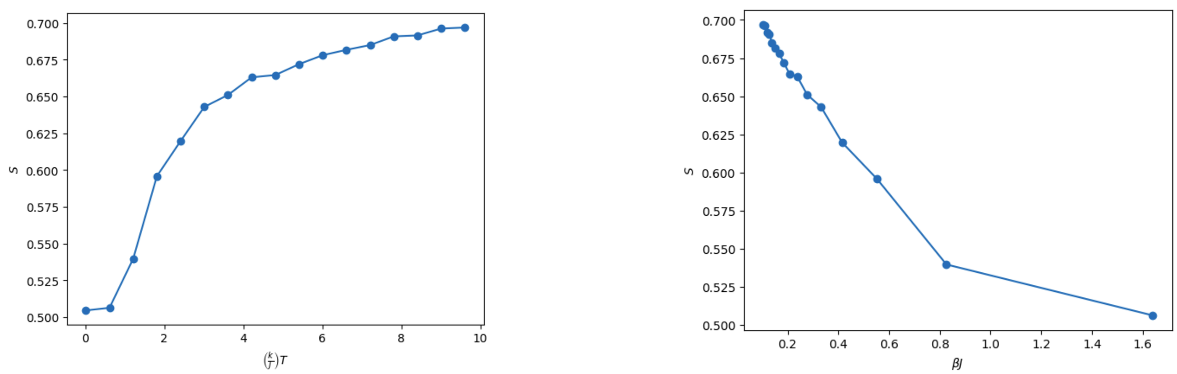


Figure 10: Entropy per site as function of T and βJ for Model C with 1st approach and N=6

Min entropy value 0.50455274 for Model C

In this case we can clearly see that the entropy is not tending to 0 even as $T \to 0$.

3.2 Approach 2: Using $q_\theta(s)$ as Generative Function

In the second approach, the model q_θ is utilized not only for probability estimation but also as a generative function. This neural network produces new lattice configurations using sampling at the end of each time node. Another ineffective way to do this is to use Metropolis algorithm to sample sequences from $q_\theta(s)$ instead of $p_\beta(s)$. But this is not needed since RNNs can be easily tweaked to generate/sample new sequences.

Following results were obtained after 200 epochs with data length (or batch size) of 64 and $L_1(\theta)$: (Orange line represents the theoretical plot)

3.2.1 2D Ising model for ferromagnetic rectangular lattice

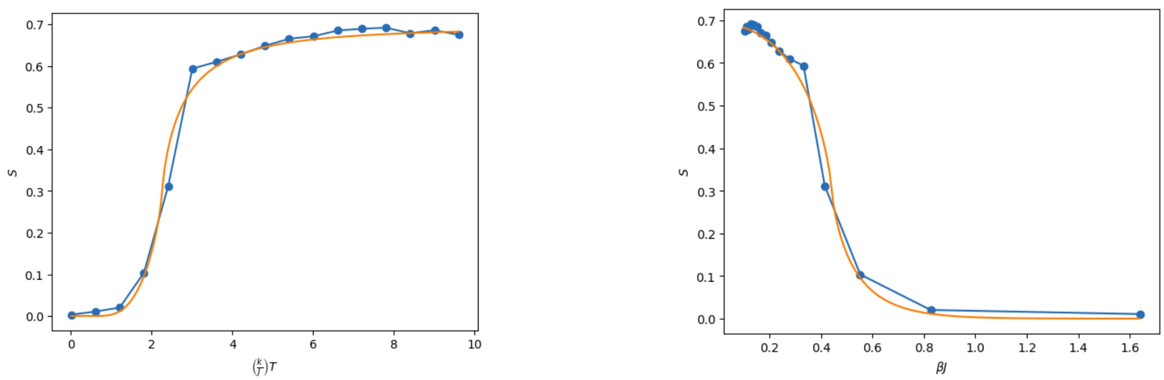


Figure 11: Entropy per site as function of T and βJ for Model A with 2nd approach and N=6

Min entropy value 0.01864889 for Model A

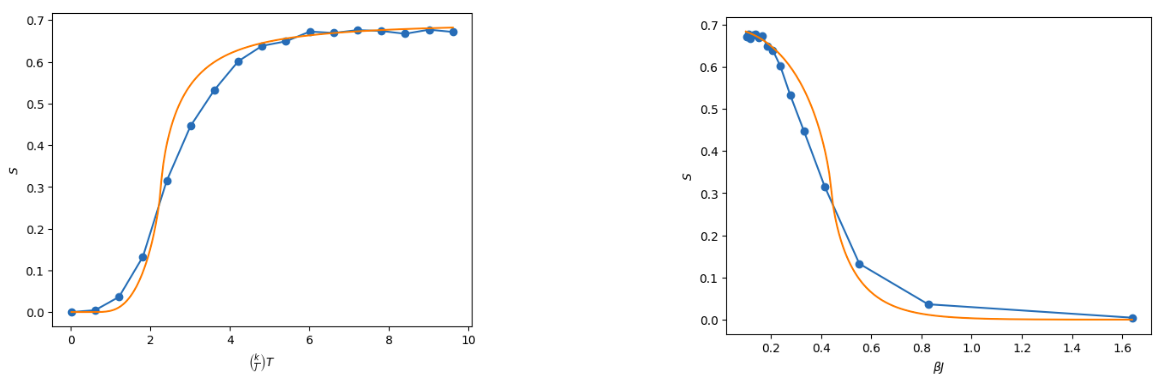


Figure 12: Entropy per site as function of T and βJ for Model C with 2nd approach and N=6

Min entropy value 1.16857449e-05 for Model C

3.2.2 2D Ising model for antiferromagnetic triangular lattice

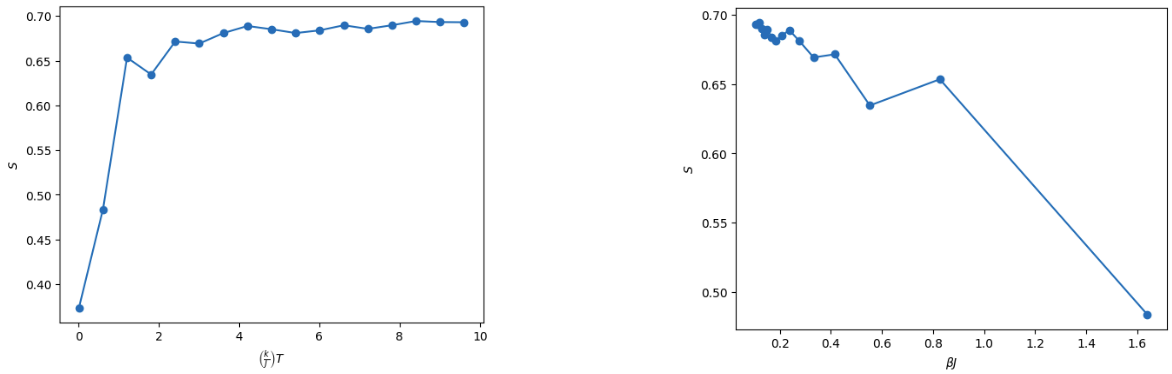


Figure 13: Entropy per site as function of T and βJ for Model A with 2nd approach and N=4

Min entropy value 0.37306831 for Model A

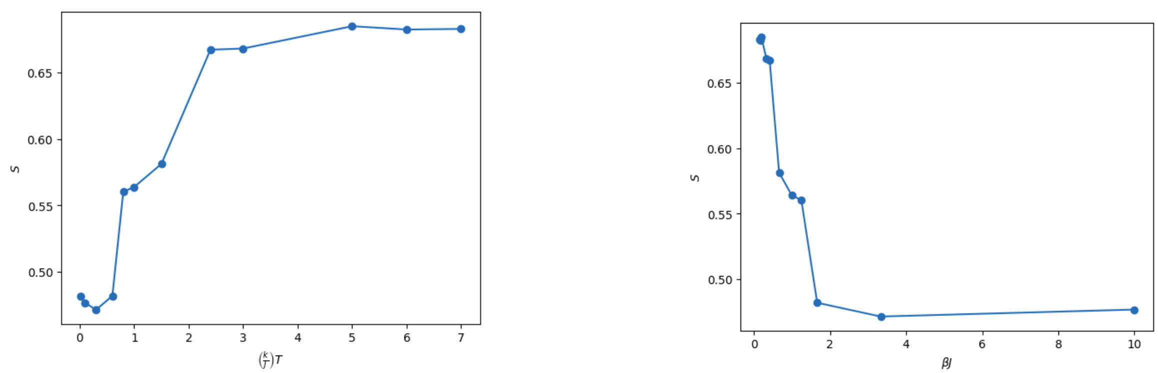


Figure 14: Entropy per site as function of T and βJ for Model A with 2nd approach and N=6 with bigger batch size 1024

Min entropy value 0.47662285 for Model A

In this case also we can clearly see that the entropy is not tending to 0 even as $T \to 0$.
Note: If the model is over trained with a small batch size then the entropy tends to 0, because the model ignores the degeneracy in the ground state and learns to generate only one of them many ground state configurations. Therefore, trainning must be done with a bigger batch size.

4 Conclusion

This project delves into the application of auto-regressive neural networks to analyze entropy as a function of temperature in ferromagnetic 2D square lattice and antiferromagnetic 2D triangular lattice systems under periodic boundary conditions. This approach opens new avenues for further exploration of complex lattice structures and make valuable contributions to the expansive domain of statistical mechanics.

References

[1] José Gaite and Denjoe O'Connor, *Field theory entropy, the H-theorem, and the renormalization group*, *Physical Review D*, vol. 54, no. 8, pp. 5163–5173, Oct 1996. http://dx.doi.org/10.1103/PhysRevD.54.5163

[2] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, The Journal of Chemical Physics, **21**(6), 1087–1092, 1953.

[3] W. K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, **57**(1), 97–109, 1970.