Ising spin configurations with the deep learning method

Yihang Zhang
Zhengzhou Foreign Language School, Zhengzhou 450001, Henan, People’s Republic of China
E-mail: shortlonglife@163.com

Keywords: Ising model, deep learning method, conditional generative adversarial network, phase transition

Abstract
In this work, we explore the statistical system’s configuration generation with generative models from Deep Learning to go beyond conventional Monte Carlo methods. Specifically, we devise a conditional generative adversarial network (cGAN) for the Ising system spin configuration generation, and we demonstrate it’s working outside the training range of temperature for the ensemble of configurations. Being different from the original GAN design, we add a further recognizer network for the constraint on the conditional parameters (in our case temperature) and also provide a diversity benefit for the generative model. We showed that the newly proposed cGAN here can learn the distribution of the Ising model for different temperatures, which can generate spin configuration efficiently with correct (within a probability distribution) temperature estimation for the microscopic configurations. Moreover, without information about criticality provided inside the training data set, it is also shown that the developed cGAN can generate Ising spin configurations around the phase transition point with the order parameter (mean magnetization) reasonably well-match to the conventional MCMC simulation but with parallel sampling advantage. We also compared typical spin configurations from cGAN with specified conditional temperature to be the critical temperature with samples simulated by MCMC, which visibly is not distinguishable. This thus can possibly help to avoid critical slowing-down as shown in the traditional Monte Carlo method.

1. Introduction
Recently the applications of machine learning techniques in scientific research have been boomed since both the currently increased computational power and the corresponding strong pattern recognition ability. Deep Learning (DL) is a branch of machine learning aiming at extracting and understanding high-level representations of data with a deeper structure of artificial neural networks [1]. After the successful development and applications from image/video processing and speech recognition, DL also showed great success in physics [2, 3], biology and engineering. It is shown that DL has the ability to effectively handling complex nonlinear systems with strong correlations which are beyond the traditional analysis ability. Tremendous progress has been made in applying deep learning techniques in condensed matter systems like classical or quantum spin models, especially in the study of phase transition identification [4, 5], and compressed quantum state representation [6], or the acceleration of Monte-Carlo simulations [7, 8]. As we know, the collective behavior of interacting degrees of freedom in the system is the core of many physical system kinds of research. Usually, the enormous number of possible free parameters that defined a near-infinite configuration space makes it super difficult to effectively model many-body systems. One thus introduces novel machine learning methods to build better approximations for the system and help to extract physics insight.

Besides these pattern recognition applications (i.e. classification and regression), there’s recently increasing interest in utilizing the generative methods for the physical system [9–11]. In general, generative models target at learning the joint probability distribution of the data for further density estimation or new sample generation. Some promising generative methods are, for example, restricted Boltzmann machine (RBM), variational autoencoder (VAE), Generative Adversarial Network (GAN), or Flow-based generative models.
It’s actually not yet well studied about the representational ability and limitations of the above mentioned generative models in physical systems. In this paper we explore Ising system spin configuration generation with conditional GAN outside of training range of temperatures, to better understand to which extent we can apply the GAN methods in many-body statistical physics for assisting or even replacing the traditional methods like MCMC or variational mean-field approaches. Besides exploring the generative model’s ability for capturing the underlying physics of the spin system, our work here also would help in compressing the physics of the system into the highly trained network to be an efficient representation, thus can largely reduce storage cost from practical consideration.

The left of the paper is structured as follows. After outlining the Ising spin model we introduce the generative adversarial network (GAN) methods. Then we develop a conditional generative adversarial network (cGAN) to be applied on Ising spin configuration generation for specified temperature, provide our numerical experiment on training and testing, discuss different aspects of the physics capturing degree for cGAN, In the end, we summarize the main finds and conclude.

2. Ising model

We consider the 2-D Ising model on the square lattice \( (N \times N \text{ grid}) \) with each site on the lattice contains one spin that points either up or down. The Ising model is a well-known model of magnetism like a ferromagnetic system of particles, for which the interactions of spins either pointing up or down \( s_i \in \{ 1, -1 \}, 1 \leq i \leq N \) is described according to the following Hamiltonian

\[
H(s) = -J \sum_{\langle i,j \rangle} s_i s_j - h \sum_i s_i,
\]

where the summation is taken over all nearest-neighbor spin pairs \( \langle i,j \rangle \) and \( h \) is an external magnetic field which is set to zero in our study here. The coupling \( J \) also called exchange energy sets the scale of interaction strength for the system. We consider \( J = 1 \) which corresponds to the ferromagnetic case and the negative sign in the Hamiltonian indicates that the spins are preferred to be parallel with each neighbor for lowering the internal energy of the system.

We consider the equilibrium Ising system here. As dictated by the ensemble theory, the probability density of a microscopic spin configuration \( s \) at equilibrium at a given temperature \( T \) is given by the Boltzmann distribution

\[
P(s|T) = \frac{1}{Z(T)} \exp\left(\frac{-H(s)}{kT}\right),
\]

where the Boltzmann constant \( k \) is set to unity in our calculation and the normalization factor \( Z \) is the partition function defined as \( Z(T) = \sum_s \exp\left(\frac{-H(s)}{kT}\right) \) with the summation running over all the configurations. Clearly one sees that such Boltzmann distribution prefers ordered states when the Ising system is at low temperature, while at high temperature this will be less required and the system would be balanced or dominated by the disorder states. From statistical physics, thermal expectations of any physical quantity \( \mathcal{O}(s) \) (e.g. energy or magnetization per degrees of freedom) that depends on the configuration \( s \) can be estimated as

\[
\langle \mathcal{O} \rangle_T = \sum_s \mathcal{O}(s) P(s|T),
\]

Numerically one usually adopts Markovian Chain Monte Carlo (MCMC) sampling to construct a set of \( N \) configurations \( s \) that are distributed according to the distribution of \( P(s|T) \) like via Metropolis-Hastings algorithm or Wolff algorithm, then under which the estimation turns out to be just \( \langle \mathcal{O}(s) \rangle_T = (1/N)\sum_i^N \mathcal{O}(s_i) \).

With lattices of size 60 \( \times \) 60, we collected Ising spin configurations sampled by Monte Carlo simulations via the Metropolis-Hastings algorithm. We prepared configurations at 40 different temperatures values from \( T = 1 \) to \( T = 3.475 \) with equal interval space in between (only 32 different temperature ensembles are taken as training set as will be explained later in training our generative model), at each temperature value 2500 configurations were generated after a sufficient equilibrium (burning time) preparation. To reduce the correlations in generating the spin configurations, the generation is with 1000 MCMC steps between two successive configurations recording on one Markov Chain.

3. Generative adversarial networks

Aimed at learning the data distribution, Generative Adversarial Network (GAN) contains two differentiable functions modeled by deep neural networks: one is the generator \( G(z) \) which maps a prior noise vector \( z \) from latent space \( (z \sim p(z)) \) to the target data space \( (\hat{x} \sim p_d(x)) \), the other one is the discriminator \( D(x) \) which tries to
distinguish real data $x$ from generated data $\hat{x} = G(z)$. The two networks will compete with each other during training, through which the generator distribution $p_G(x)$ can be pulled to approach the underlying data distribution $p_{\text{true}}(x)$.

The vanilla generative adversarial networks, GAN, involves two loss functions $\mathcal{L}_D$ and $\mathcal{L}_G$ for the discriminator and the generator, respectively. By mimicking a zero-sum game with $\mathcal{L}_G = -\mathcal{L}_D$, GAN consists of optimization for the respective network’s parameters $\theta_G$ and $\theta_D$ to be converged in the game to

$$
\theta_{G,D}^* = \arg \min_{\theta_G} \max_{\theta_D} (-\mathcal{L}_D(\theta_G, \theta_D)).
$$

(4)

In the original GAN, the binary cross-entropy loss function was used,

$$
\mathcal{L}_D = -\mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] - \mathbb{E}_{z \sim p_{z}} [\log (1 - D(G(z)))]
$$

(5)

with

$$
\mathbb{E}_{x \sim p}[A] = \int dx p(x)A(x), \quad \int dx p(x) = 1
$$

(6)

represents the expected value over the normalized probability distribution $p(x)$ and one used

$$
\mathbb{E}_{z \sim p_{z}} [A(G(z))] = \mathbb{E}_{z \sim p_{z}} [A(\hat{x})].
$$

(7)

Note that for the generator, the first term of (5) has no influence on generator $G$ during training as it only depends on $\theta_D$. The expectation values in the above loss functions are evaluated in training from the mean of all the training samples. $\theta_D$ and $\theta_G$ are updated via backpropagation with the gradients of the loss functions,

$$
\sum_i \nabla_{\theta_D} \mathcal{L}_D(x_i, z_i; \theta_D), \quad \sum_i \nabla_{\theta_G} \mathcal{L}_G(z_i; \theta_G),
$$

(8)

with $x_i$ the sample from the training set and $z_i$ the latent noise is drawn from the prior distribution $p_{\text{prior}}$. With the above binary cross-entropy loss function, the optimal discriminator for given fixed generator $G$ can be derived to be

$$
\mathcal{D}_G^*(x) = \frac{p_{\text{true}}(x)}{p_{\text{true}}(x) + p_G(x)}.
$$

(9)

Taking the information theory point of view, the above training objective for discriminator (thus the training criterion of the generator) are just the Jensen–Shannon (JS) divergence, which is a measure of similarity between two probability distributions,

$$
\max_{\theta_D} (-\mathcal{L}_D(\theta_G, \theta_D)) = -\mathcal{L}_D^*
$$

(10)

$$
= 2 \mathcal{D}_{\text{JS}}(p_{\text{true}}\|p_G) - 2 \log 2,
$$

(11)

where the JS divergence is formulated by symmetrizing the Kullback-Leibler (KL) divergence,

$$
\mathcal{D}_{\text{JS}}(p\|q) = \frac{1}{2} \left[ \mathcal{D}_{\text{KL}}(p\|\frac{p+q}{2}) + \mathcal{D}_{\text{KL}}(q\|\frac{p+q}{2}) \right],
$$

(12)

with the KL divergence defined as

$$
\mathcal{D}_{\text{KL}}(p\|q) = \int dx p(x) \log \frac{p(x)}{q(x)}.
$$

(13)

So the optima for the generator is reached if and only if $p_G(x) = p_{\text{true}}(x)$, resulting in $D^* = 1/2$ for the global optimum of the zero-sum (minimax) game which is also called Nash-equilibrium.

Practically it’s very hard to train with this default GAN setup especially for high-dimensional cases. With only the above JS divergence measure, the discriminator $D(x)$ may not provide sufficient information to measure the distance between the generated distribution $p_G$ and the real data distribution $p_{\text{true}}$ when the two distributions do not have any overlap. Mathematically, when the support of $p_G$ and $p_{\text{true}}$ both rest in low dimensional manifolds of the data space, the two distributions thus have a zero measure overlap which results in a vanishing gradient for the generator. This leads to a weak training signal for $G$ updating and also general instability. Mode-collapse can easily occur for GAN where the generator only learns to produce a single element (mode) in the state space that can maximally confuse the discriminator. In order to avoid this kind of failure training, a multitude of different techniques has been developed recently, like ACGAN [12], WGAN [13], improved WGAN, which help to stabilize and improve the GAN training. We used the improved WGAN with gradient penalty [14] in this work. The most important difference of WGAN compared to the original GAN lies in the loss function, where the Wasserstein-distance (also called Earth Mover distance) provides an efficient measure for the distance between the two distributions ($p_{\text{true}}$ and $p_G$) even if they are not overlapping anywhere. The loss functions are now
\[ \mathcal{L}_D = -\mathbb{E}_{x \sim p_{\text{data}}} [D(x)] + \mathbb{E}_{z \sim p_{\text{prior}}}[D(G(z))] + \lambda \mathbb{E}_{z \sim p_{\text{prior}}} [\| \nabla_x D(x) \|_p - K]^2, \]  

(14)

and

\[ \mathcal{L}_G = -\mathbb{E}_{z \sim p_{\text{prior}}}[D(G(z))], \]  

(15)

where the gradient penalty term with strength \( \lambda \) is computed in a linearly interpolated sample space,

\[ \hat{x}_{\epsilon p} = \epsilon x + (1 - \epsilon) \hat{x}, \]  

(16)

with uniformly sampled \( \epsilon \sim (0, 1) \).

### 4. Ising configuration generation with conditional GAN

With the Gradient Penalty improved Wasserstein GAN (WGAN-gp) as the basic adversarial training framework, here we build a conditional GAN (cGAN) for generating Ising spin configurations at conditionally specified temperature values.

#### 4.1. cGAN architecture

Figure 1 depicted the cGAN structure developed in the present study. It consists of three models: a generator \( G(z, T) \) for spin configuration generation conditioned on temperature \( T \) with prior source \( z \sim p(z) \), a discriminator \( D(x, T) \) for distinguishing the spin configurations by giving a Wasserstein distance measure and a recognizer \( R(x) \) for estimating the temperature associated with the spin configuration.

As explored in [15], for the input of the discriminator the condition of temperature can be added as a second channel of the spin configurations. Inside the generator, we propose to add the condition of temperature as a scale factor for the latent code \( z \sim p(z) \) which works better than other embedding methods from our experiments. As a further push for diversity requirement of the generated distribution, we additionally consider maximizing the entropy for the generator’s distribution which turns out to be a reconstruction error for an additional recognizer network for the condition of temperature,

\[ \mathcal{L}_R = -\mathbb{E}_{x \sim p_{\text{data}}} [\log p_{\text{th}}(T|x)], \]  

(17)

by assuming the distribution modeled by the recognizer \( R \) to be gaussian one immediately obtain the normal Euclidean distance between the model prediction and ground-truth.

#### 4.2. Training and testing results

As our main focus, we explore here the generalizability of the trained cGAN in generating spin configuration outside of the training set range for the temperature. Specifically, we excluded in the training set spin configuration ensembles with the temperature around the vicinity of the phase transition. This critical region is the most time-consuming part for conventional MCMC simulation since the auto-correlation time will diverge while approaching the critical point (which is also called critical slowing-down). The training set of configuration ensembles cover temperature interval \([1.0, 2.0] \cup [2.5, 3.5]\) from MCMC simulations.

After training, we purposely specify conditional temperature values in the critical range \([2.0, 2.5]\) for the cGAN generation, and further test the recognizer prediction (i.e. temperature evaluation) on the cGAN generated configurations and MCMC generated training configurations. The results are shown in figure 2.
We see that the recognizer $R(x)$ successfully learned the temperature estimation for each individual spin configuration. Note that from statistical theory, under a fixed temperature every spin configuration has a probability to appear according to equation (2), so the temperature is associated with a randomly chosen spin configuration should also follow a distribution but not a deterministic value. That’s why in figure 2 the network predicted temperature spread with the center in the ground truth value, even though in the training for each MC generated spin configurations we label them with the specific temperature value during MCMC generation. For the cGAN generated configurations at unseen critical temperature region, the recognizer also consistently predict the correct temperature estimation for them (see the temperature range $[2.0, 2.5]$ in figure 2). We stress that for the training only temperature range $[1.0, 2.0] \cup [2.5, 3.5]$ was provided, but the agreement between the desired conditional temperature values and the supervised trained prediction for temperature on the generated configurations is spectacular over a much broader range of temperature values.

We take a closer look at the microscopic configurations generated from the trained cGAN with conditional temperature specified to be the critical temperature $T = T_c$, to check that if the trained generative model here can capture the criticality physics based on training configurations without any criticality information. Physically speaking, it’s also quite interesting to know how much criticality information can be contained inside configuration ensembles away from the critical region. In principle only when the ensemble approach infinity (to be large enough to demonstrate the Ergodic hypothesis), all the physics of the system can be represented which actually is the physics inside the partition function of the system. Here, taking the approximation point of view that the MC generated configuration ensembles can provide numerical construction for the partition function for further observable estimation, we can imagine that the cGAN can first learn the interpolation of the partition function in temperature and then reverse it back to the microscopic configuration space. One typical configuration generated by cGAN at the critical temperature (which is not included in the training data set) and with magnetization around the mean value is shown in figure 3. Since the continuous numerical output nature of the designed cGAN network, there are some shining ‘star’ points appeared to which one can use around the operation to get rid of easily and would not change the thermal dynamics observable evaluation visibly.

We then evaluated the mean magnetization per lattice site for the Ising configurations ensemble at different temperatures with the trained cGAN, and compare it with the MCMC results in figure 4, where 100 configurations are generated to evaluate the mean magnetization for each temperature value by the trained cGAN. To take into account the model uncertainty approximately, we save three different trained cGAN version by stopping the training at different epochs. From figure 4 we see that the general temperature dependency of magnetization for configurations is represented well by the generator in cGAN. We attempt to use the network to generalize the distribution that it was trained on, so most interestingly are the critical region that is not included in the training for the cGAN. Clearly, we see the network got the ability to make this high-dimensional interpolation for configuration generation at conditional temperature values outside of the training range, which reasonably captured the average order parameter observable, mean magnetization.
Figure 3. Typical configuration generated from the trained cGAN at specified temperature $T = T_c$ (upper and middle panel, with the middle one, take a round operation for the generated configuration) and from MCMC (bottom panel), note that the training set didn’t cover the temperature around the critical range as explained in the text.

Figure 4. The mean magnetization per site on the configurations generated by the cGAN and MCMC versus temperature. ‘cGAN1’, ‘cGAN2’ and ‘cGAN3’ represent three different versions of the cGAN from training. Note that the training set covers only the temperature range $[1.0, 2.0] \cup [2.5, 3.5]$. 
5. Conclusions

In this paper, we developed a conditional GAN for Ising spin configuration generation under specified temperature values and successfully trained it for configurations generation with conditional temperature values outside of the training set. The Ising spin configurations generated from our generator well replicate the underlying physical distributions of the training set, also the generator in the cGAN well captured the high dimensional correlations between the microscopic spin configurations and temperature on an ensemble average manner. Most interestingly, the criticality information which is not provided in the training set a be partly revealed by the well-trained cGAN, for which we checked the order parameter estimation in the critical region that agrees reasonably well with the Monte Carlo simulation. This can be helpful in handling critical slowing-down near criticality as in MCMC simulations like to be taken as an uncorrelated proposal in the Markov Chain. It can also be useful in compressing information of the studies physical system which in our case Ising systems. In the future, we will test more physics like spin-spin correlations comparisons. We will also take into account the boundary condition and symmetries of the system explicitly into the Generator.

ORCID iDs

Yihang Zhang  © https://orcid.org/0000-0002-4931-7127

References

[1] LeCun Y, Bengio Y and Hinton G. 2015 Nature 521 436–44
[2] Baldi P, Sadowski P and Whiteon D 2014 Nature Commun. 5 4308
[3] Pang L G, Zhou K, Su N, Petersen H, Stöcker H and Wang X N 2018 Nature Commun 9 210
[4] van Nieuwenburg E P L, Liu Y H and Huber S D 2017 Nat. Phys. 13 435–9
[5] Carraquilla J and Melko R G 2017 Nat. Phys. 13 431–4
[6] Carleo G and Troyer M 2017 Science 355 602–6
[7] Xu X Y, Qi Y, Liu J, Fu L and Meng Z Y 2017 Phys. Rev. B 96 041119
[8] Torlai G and Melko R G 2016 Phys Rev. B 94 165134
[9] Zhou K, Endrödi G, Pang L G and Stöcker H 2019 Phys. Rev. D 100 011501
[10] Shanahan P E, Trewartha D and Detmold W 2018 Phys. Rev. D 97 094506
[11] Wang L, Jiang Y, He L and Zhou K arXiv:2005.04857 [cond-mat.dis-nn].
[12] Odena A, Olah C and Shlens J arXiv:1610.09585
[13] Arjovsky M, Chintala S and Bottou L arXiv:1701.07875
[14] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V and Courville A arXiv:1704.00028
[15] Liu Z, Rodrigues Sean P and Cai W arXiv:1710.04987