Variational Autoencoders for Classical Spin Models

Benjamin Nosarzewski
Stanford University
bln@stanford.edu

Abstract

Lattice spin models are used to study the magnetic behavior of interacting electrons in solids. We study the one and two dimensional nearest-neighbor Ising model and the one dimensional XY model to demonstrate that variational autoencoders with convolutional layers can learn to properly sample from the correct distribution of spin configurations across a range of temperatures and capture a phase transition. Generative neural networks produce uncorrelated samples and therefore offer an advantage over Markov chain Monte Carlo methods which often suffer from long autocorrelation times.

1. Introduction

An understanding of emergent behavior in many-body systems is a great challenge that has been the subject of intense research in the physics community for many decades and has lead to the invention and discovery of many novel materials with practical technological applications in the real world [1, 2, 3, 4, 5]. Theoretical approaches to this problem often utilize simplified models to represent the interactions involving electrons and the lattice in real materials. Spin models are one particular type of idealized model used to study magnetism. In these models the kinetic energy of electrons hopping on a lattice is ignored and only the interactions between the electron spins, a quantity proportional to the magnetic moment, is considered. For classical (non-quantum mechanical) models, the probability of a given spin configuration is proportional to the Boltzmann weight which depends on the energy of the state and the temperature of the system and is straightforwardly computed for a given configuration [6]. The temperature of the system determines the strength of thermal fluctuations which can drive the system through phase transitions. To illustrate the importance of such models, the 2017 Nobel Prize in physics was awarded in part for the understanding the topological phase transition which occurs in the two dimensional XY spin model [7, 8].

Even the simplest case of the nearest-neighbor Ising model on a square lattice has only been solved analytically in one and two dimensions [9]. The general intractability of the many-body problem by exact analytic methods has lead to the development of a wide variety of numerical techniques such as exact diagonalization, dynamical mean-field theory, density matrix renormalization group, and Monte Carlo based methods [10, 11, 6, 12]. More recently, restricted Boltzmann machines (RBMs) have also been successfully applied to solving Ising-like models [13, 14, 15]. However, neural network architectures such as variational autoencoders (VAEs) and generative adversarial networks (GANs) have been more successful as generative models than the somewhat outdated RBMs [16, 17]. We demonstrate that VAEs and the recently developed method of adversarial variational Bayes (AVB), which combines elements of VAEs and GANs, can be used to solve classical spin models in one and two dimensions [18]. Our neural networks are successful even across a phase transition and capture the average energy, spin-spin correlation function, and overall distribution of states. In addition, generative neural networks provide uncorrelated samples unlike Markov Chain Monte Carlo methods which often suffer from long autocorrelation times [19, 16].

The techniques developed in the past decade for the application of deep neural networks to computer vision problems, namely the use of convolutional layers [20], are particularly useful for neural networks trained to solve spin models. This is because the Hamiltonian which describes these systems typically has translation invariance which stems from the fact that an idealized solid looks the same from any one of its lattice sites. In other words, patterns in the spin configurations can be translated in space without changing the energy of the configuration. Therefore spin configurations can be viewed as images where each spin corresponds to a pixel which can either take on discrete or continuous values depending on the model. The goal of the neural network is then to learn the likelihood and structure of important features in the spin configurations which can occur with equal probability anywhere in the system, in analogy to features in real images. As expected, we find that convolutional layers are able to improve the performance of

the network, likely because they are able to successfully detect the occurence of domain wall features (spin flips) in the configurations. In addition, the realization that some deep neural networks perform the same coarse-graining procedure as the powerful renormalization group method in theoretical physics suggests that such networks are promising tools for studying many-body systems [15].

We study the classical Ising model in one and two dimensions and the classical XY model in one dimension. The configuration energy for the Ising model is given by:

$$H = -\sum_{\langle ij \rangle} J\sigma_i\sigma_j \tag{1}$$

where each spin variable σ takes on a discrete value of -1 or 1 and $\langle ij \rangle$ denotes nearest-neighbor pairs on the lattice. The configuration energy for the XY model is given by:

$$H = -\sum_{\langle ij \rangle} J\mathbf{s}_i \cdot \mathbf{s}_j = -\sum_{\langle ij \rangle} J\cos(\theta_i - \theta_j) \tag{2}$$

where each spin is represented by an two-dimensional unit-length vector $\mathbf{s} = (\cos\theta, \sin\theta)$ with $\theta \in [-\pi, \pi]$. The probability of a given spin configuration with energy E at temperature T is proportional to the Boltzmann weight $\exp(-E/T)$ in units where the Boltzmann constant $k_B = 1$. The metrics we use to evaluate our results include the average energy, spin-spin correlation function, and the distribution of states in energy (density of states). The spin-spin correlation function is obtained by evaluating the expectation value $\langle \mathbf{s}_i \mathbf{s}_j \rangle$.

2. Methods

The networks are implemented in Tensorflow and are trained on an Nvidia Tesla K80 GPU using Google Cloud compute resources.

2.1. Training data

Training data is obtained from Monte Carlo performed with the Metropolis-Hastings algorithm [6]. For each Markov chain, the procedure begins with a random spin configuration and then generates subsequent configurations via importance sampling. For each Monte Carlo step a random lattice site is chosen and an update to its spin is proposed. The energy difference ΔE between the proposed state and the current state is calculated, which only involves the spins of the nearest neighbors for the models we consider, and the proposed state is accepted with probability $r = \exp(-\Delta E/T)$. Our Monte Carlo data is generated using 20 independent Markov Chains for each temperature in the simulation. Each chain is generated after performing 2000 warm-up sweeps followed by 10,000 sweeps with measurements taken every 20 sweeps. The training set for each temperature therefore consists of 10,000 spin configurations. We then train a VAE or AVB on each of the training sets and use it to generate 10,000 random samples which we analyze and use to compute the expectation values of the desired observables.

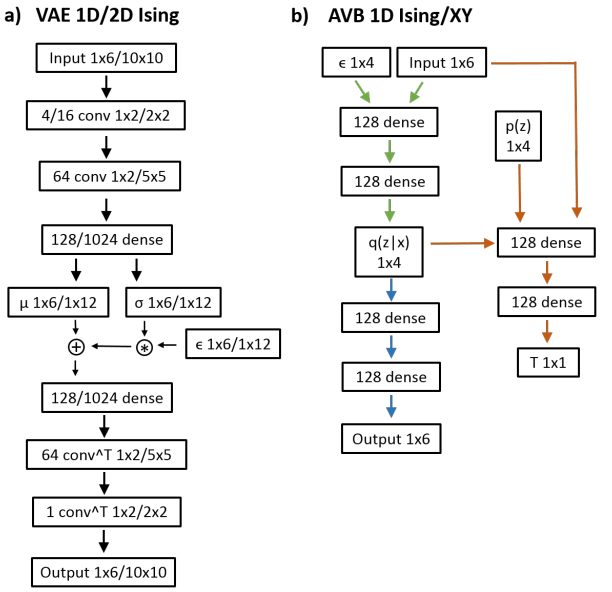

Figure 1. **Architectures.** a) The variational autoencoder architectures used for 1D and 2D Ising models. b) The adversarial variational Bayes architectures used for the 1D Ising and XY models.

2.2. Variational autoencoder

A variation autoencoder (VAE) resembles a traditional autoencoder in that it has an encoder and a decoder and attempts to minimize the reconstruction loss between the input and the output [21]. However, during training an additional term in the loss function reflects that the prior on the latent variable z generated by the encoder is chosen to be normally distributed: $p(z) = N(0, I)$. The encoder of the VAE attempts to learn the posterior distribution $p(z|X)$ so that it can sample values of z which are likely to correspond to an input X. At test time, the encoder is discarded and new samples are generated by using Gaussian random noise (the chosen prior) for the hidden variable z as input to the decoder.

The translation invariance of the system suggests that convolutional layers are well-suited for the problem, and we have achieved the best performance with our VAE by using convolutional layers for the encoding step and deconvolutional layers for the decoding step. The specific parameters in the architecture are outlined in Fig. 1a for both one and two dimensional cases. The ϵ in Fig. 1a is a random normally distributed variable used for the reparametrization trick. The loss function for the network is given by:

$$L(\theta,\phi;x) = \frac{1}{N}\sum_i D_{KL}(q(z|x_i)||N(0,I)) - \log p(x_i)$$
$$= \frac{1}{N}\sum_i D_{KL}(N(\mu,\sigma)||N(0,I)) + |x_i - y_i|^2 \tag{3}$$

where D_{KL} is the KL divergence, $N(0,I)$ is a unit normal Gaussian distribution, y_i is the reconstructed output for sample x_i, ϕ represents the parameters for the encoder $q(z|x)$, θ represents the parameters for the decoder $p(x)$, μ and σ are the latent variables produced by the encoder, and N is the batch size. The KL divergence penalty encourages the distribution for the latent variable z produced by the encoder $q(z|x)$ to be Gaussian.

We found that regularization such as L_2 weight regularization, dropout, and batch normalization were not helpful in the model, possibly because sufficient regularization is already provided by the KL divergence term in the loss function and the random noise used to sample from $q(z|x)$. We used the Adam optimizer for training with a learning rate of 0.001. Since variational autoencoders perform better with larger batch sizes which provide a better representation of the data distribution, we used a batch size of 500 when training all of our models.

2.3. Adversarial variational Bayes

We also implement the recently developed technique of adversarial variational Bayes (AVB) which allows for more expressive inference models by introducing an auxiliary discriminative network in addition to the traditional VAE architecture [18]. In a simple test example on a synthetic dataset consisting of four classes of binary images, AVB partitions the distribution of the hidden variable more symmetrically for the four classes than VAE and also achieves an improved log-likehood score and less reconstruction error [18]. The optimization objective for AVB is derived starting from the VAE objective and expressing the KL divergence as the function which maximizes the objective of a secondary, discriminative network:

$$\max_T [E_{p_D(x)} E_{q_\phi(z|x)} \log\sigma(T(x,z)) + E_{p_D(x)} E_{p(z)} \log(1-\sigma(T(x,z)))] \tag{4}$$

where $p_D(x)$ is the data distribution, σ is a sigmoid, $p(z)$ is the prior, $q_\phi(z|x)$ the inference network, and T the output of the discriminative network. Intuitively, $T(x,z)$ tries to distinguish pairs (x,z) sampled from $p_D(x)p(z)$ and $p_D(x)q_\phi(z|x)$. The original objective then becomes:

$$\max_{\theta,\phi} E_{p_D(x)} E_{q_\phi(z|x)}[-T^*(x,z)) + \log p_\theta(x|z)] \tag{5}$$

where T^* is the maximal discriminator according to (4). It can be proven that $E_{q_\phi(z|x)}T^*(x,z) = D_{KL}(q_\phi(z|x),p(z))$ which establishes the connection to the original VAE objective. Therefore training the network involves alternating gradient ascent updates according to the objectives given by (4) and (5).

As shown in Fig. 1b, we have only used dense (fully-connected) layers in our AVB so far. We used the Adam optimizer for training with a learning rate of 0.002 and $\beta_1 = 0.4$. These networks are more difficult to train than traditional VAEs because the success of the training is difficult to determine based on the behavior of loss function versus training epoch alone. In the future we would like to experiment with more complicated architectures and convolutional layers.

2.4. Analytic solutions

We compare the average energy predicted by our networks to the analytic result which can be obtained from the partition function as follows:

$$\langle E\rangle = -\frac{\partial \ln Z}{\partial\beta} \tag{6}$$

where β is the inverse temperature. The partition functions for the spin models we consider are known analytically. For the 1D Ising model with free boundary conditions, the partition function is:

$$Z = 2(2\cosh(\beta J))^{N-1} \tag{7}$$

where N is the number of sites [9]. For the 1D Ising model we also know the spin correlation function is: $\langle\sigma_i\sigma_{i+N}\rangle = \tanh(\beta J)^N$. For the 2D Ising model with periodic boundary conditions, the partition function is given by:

$$\begin{aligned} Z &= \lambda^N \\ \ln\lambda &= \ln(2\cosh(2\beta J)) \\ &\quad + \frac{1}{\pi}\int_0^{\pi/2} dw \ln\left[\frac{1}{2}(1+\sqrt{1-K^2\sin^2(w)})\right] \\ K &= \frac{2\sinh(2\beta J)}{(\cosh(2\beta J))^2}. \end{aligned} \tag{8}$$

The ferromagnetic phase transition occurs at a temperature given by $T_c = \frac{2J}{\ln(1+\sqrt{2})} \approx 2.27J$ [9]. Finally, for the 1D XY model the partition function is:

$$Z = (2\pi)^N I_0(\beta J)^{N-1} \tag{9}$$

where I_0 is the modified Bessel function of the first kind [22]. We demonstrate that our Monte Carlo data of course reproduces the average energy versus temperature within error bars as shown by the blue markers in Fig. 2 and we are therefore confident that it represents the true distribution.

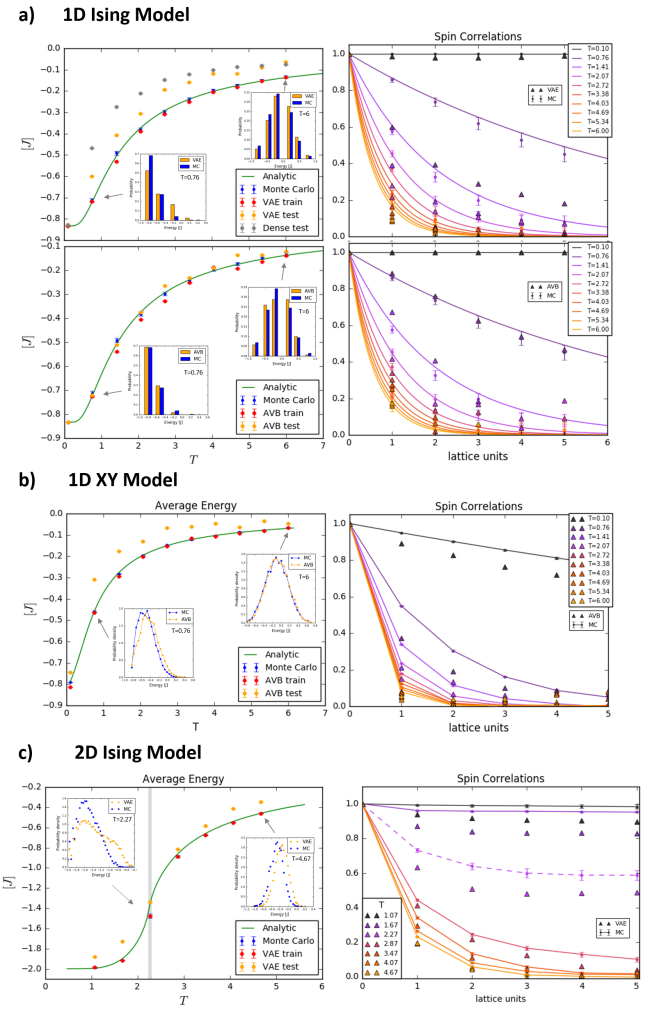

Figure 2. **Results.** a) 1D Ising model with 6 sites using VAE and AVB. b) 1D XY model with 6 sites using AVB. c) 2D Ising model on a 10x10 lattice with VAE.

We use the Monte Carlo data as a baseline for the density of states and spin correlations in the cases where analytic solutions are unavailable.

3. Results

The performance of our neural networks is presented in Fig. 2. In the leftmost panels the solid green line corresponds to the analytic solution for the average energy per site. In each case, the trends for average energy and the spin correlation function versus temperature are captured by the networks, in some cases better than others. For the 1D Ising model shown in Fig. 2a, the AVB network in the upper panel outperforms the VAE in the panel below in terms of accurately capturing the average energy and especially the spin correlations. Note that the VAE significantly underestimates the spin correlations at the second lowest temperature

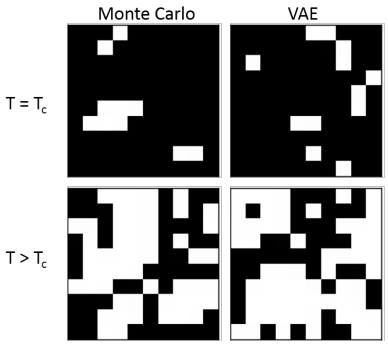

Figure 3. **Visualization of states.** Representative samples draw with energy equal to average energy for Monte Carlo and the variational autoencoder for the 2D Ising model at the critical temperature and above the critical temperature. Black and white pixels correspond to spin up and down on the 10x10 lattice.

in the simulations while AVB does not. This is a promising result which suggests that extending our AVB networks to two dimensions may further improve results in those cases. Insets in the leftmost panels display histograms of the density of states at selected temperatures which are generated from the neural networks and compared to the true distribution obtained from Monte Carlo. The distribution of states is particularly well captured in the 1D XY model in Fig. 2b which demonstrates that our methods apply equally well to systems with continuous spin variables. In the two dimensional Ising model, the VAE successfully captures the average energy versus temperature even across the critical temperature for the ferromagnetic phase transition as shown in Fig. 2c. Note that in every case the networks perform nearly perfectly at training time (see red markers in leftmost panels of Fig. 2) which suggests that there may be some over-fitting.

In Fig. 3 we visualize some sample states generated by the VAE for the 2D Ising model and those generated by Monte Carlo to perform a qualitative comparison. We checked that these particular Monte Carlo samples are reconstructed perfectly when passed through the network, which is not surprising given the high accuracy of the networks at training time. The Monte Carlo samples are selected samples with energy equal to the average energy predicted by Monte Carlo, and the VAE samples are selected samples with energy equal to the average energy predicted by the VAE. We see qualitative agreement between the samples at the transition temperature as well as above the transition temperature. This verifies that despite the small discrepancies in the average energy predicted between Monte Carlo and the neural networks, the spin configurations are qualitatively similar. As expected, compared to the high temperature spin configurations, the spin configurations at the transition temperature contain far fewer

spin flips (which come with an energy cost) and demonstrate a tendency towards ferromagnetic ordering where all spins tend to point in one direction. At higher temperatures thermal fluctuations lead to an increased number of spin flips resulting in a more random, uncorrelated spin pattern. The location and frequency of spin flips determines the energy of the configuration as well as other properties such as the spin correlations. Note that spin flips simply correspond to edges in the images of the spin configurations visualized in Fig 3. Since convolutional layers are designed to detect features such as edges, we argue that the ability of convolutional layers to detect spin flips is responsible for the improvement we see over dense (fully-connected) networks. An example of the quantitative improvement from use of convolutional layers in our VAE is shown in the top-left panel of Fig. 2a where a VAE with convolutional layers performs better than a VAE with only dense layers for which the results are plotted with the gray markers.

4. Conclusion

It is exciting that neural networks are able to learn the true distribution of spin configurations for both discrete and continuous spin models. Furthermore, if used cleverly, generative neural networks which produce uncorrelated samples may provide a computational advantage over Markov Chain Monte Carlo methods which often suffer from long autocorrelation times. Techniques such as amortized Monte Carlo could be used to train these networks efficiently [19]. We would like to apply VAEs and AVBs to more interesting spin models such as the 2D XY model which exhibits a vortex-unbinding phase transition, models with frustrated magnetism, and models with quantum dynamics. Our work demonstrates that deep learning could potentially prove to be a useful computational technique in condensed matter physics.

5. Acknowledgements

We thank Gabriel Maher and Ben Poole for helpful discussion and for suggesting variational autoencoders and adversarial variational Bayes. We also thank course staff for CS231n for providing funding for compute resources on the Google Cloud Platform.

References

[1] M. Riordan, L. Hodesson, and C. Herring, The Invention of the Transistor, Rev. Modern Phys., 71, SS336-S345, 1999

[2] K. Onnes, The resistance of pure mercury at helium temperatures, Comm. Phys. Lab. Univ. Leiden; No. 120b, 1911

[3] P. Anderson, Twenty-five Years of High-Temperature Superconductivity - A Personal Review, Journal of Physics: Conference Series 449, 2013

[4] M. Hasan and J. Moore, Three-Dimensional Topological Insulators, Vol. 2:55-78, 2011

[5] J. Bardeen, L. Cooper, and J. Schrieffer, Theory of Superconductivity, Vol. 108, 5, 1957

[6] R. Santos, Introduction to Quantum Monte Carlo Simulations for Fermionic Systems, Brazilian Journal of Physics, vol. 33, no. 1, 2003

[7] J. Kosterlitz, The critical properties of the two-dimensional xy model. Journal of PHysics C: Solid State Physics, 7(6):1046, 1974

[8] J. Kosterlitz and D. Thouless, Long range order and metastability in two dimensional solids and superfluids.(Application of dislocation theory). Journal of Physics C: Solid State Physics, 5(11):L124, 1972

[9] W. Cai, Introduction to Statistical Mechanics, 2011

[10] Numerical Methods in Quantum Condensed Matter and Lattice Gauge Systems, Proseminar in Theoretical Physics, ETH Honggerberg, 2005

[11] S. White, Density Matrix formulation for quantum renormalization groups, Phys. Rev. Lett. 69, 2863, 1992

[12] N. Kawashima and K. Harada, Recent Developments of World-Line Monte Carlo Methods, arXiv:cond-mat/0312675v2, 2004

[13] A. Aoki and T. Kobayashi, Restricted Boltzmann Machines for the Long Range Ising Models, arXiv:1701.00246,2017

[14] L. Wang, Can Boltzmann Machines Discover Cluster Updates?, arXiv:1702.08586, 2017

[15] P. Mehta and D. J. Schwab, An exact mapping between the variational renormalization group and deep learning, arXiv: 1410.3831 2014

[16] D. Kingma and M. Welling, Auto-Encoding Variational Bayes, arXiv:1312.6114, 2014

[17] I. Goodfellow, J. Puget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative Adversairal Networks, arXiv:1406.2661, 2014

[18] L. Mescheder, S. Nowozin, A. Geiger, Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks, arXiv:1701.04722, 2017

[19] Y. Li, R. Turner, and Q. Liu, Approximate Inference with Amortised MCMC, arXiv:1702.08343, 2017

[20] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems 25 (NIPS), 2012

[21] Carl Doersch, Tutorial on Variational Autoencoders, arXiv:1606.05908, 2016

[22] I. Vilfan, Statistical Mechanics, Ch 2, http://www-f1.ijs.si/ vilfan/SM/