Week 2

Progress during 6-12th July

Hao SUN

July 12, 2017

Contents

About this project

- It's Edward's idea to use unsupervised learning to perform star galaxy classification.[Github]
- Method used: surrogate class + CNNs + hypercolumn
- Problem: too many surrogate classes

MCMC scale expansion: Motivation

- Implementing large scale MCMC is time consuming
- What MCMC does in generating different spin configurations is flipping the spins to satisfy the Ergodic Hypothesis. Which makes MCMC a NP-hard problem.
- Generative models, such as AEs, VAEs and GANs, are promising in generating scientific datas.

Previous research on ML+Ising

1. [1410.3831v1]An exact mapping between the Variational Renormalization Group and Deep Learning. The author showed the relationship between renormalization group(RG) and Restricted Boltzmann Machines(RBM) in Ising model.
2. [1606.00318]Discovering phase transitions with unsupervised learning. The author used principal component analysis(PCA) to find order parameters of Ising model.
3. [1605.01735]Machine learning phases of matter. The author identified phases and phase transitions in condensed matter systems via supervised machine learning.
4. [1703.02435]Unsupervised learning of phase transitions: from principal component analysis to variational autoencoders. The author used PCA, AE and VAE to learn about the latent parameters which best describe states of the two-dimensional Ising model and the three-dimensional XY model.

Structure of AE/VAEs

Input datas are generated from MCMC simulation. At each temperature [0.0, 5.0], I have 400 different spin configurations saved in a numpy array. So the input shape of my data is (50 * 400, 256). Where the 256=16*16 shows the size of my lattice.

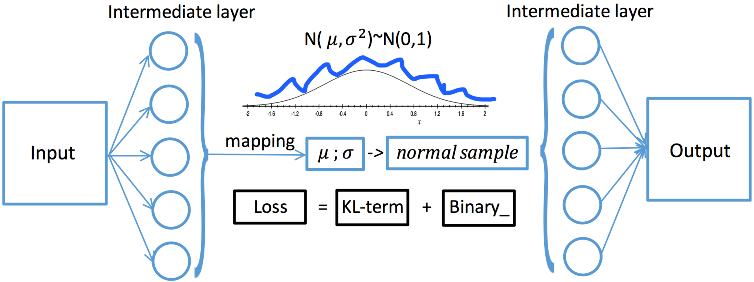


Figure: Structure of VAE(AE)

Duplicate the main results in 1703.02435

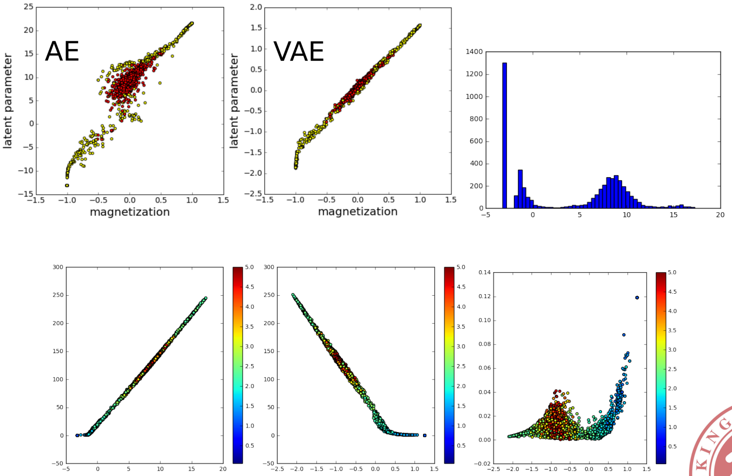
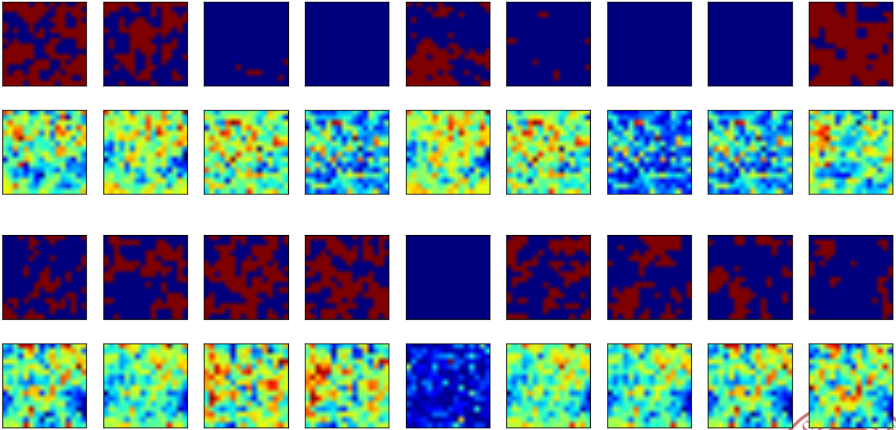


Figure: AE/VAE/Hist-Encoded & VAE-Var

Spin configurations generated



Red points: spin up(+1), blue points: spin down(0)

Problem:

1. away from spin +1 and 0, but around 0.5 for most spin units
2. mode collapse

Revise the loss function

- the loss function now was

$$Loss = KLD + L_2 norm$$

  This loss function works well if the goal is to reduce the dimension into two or three and separate different clusters in this 2or3-D space easily.
- However, spin configurations have the same magnetic moment don't have to have the same spin value at each point, as the loss function demands.
- e.g. when magnetic moment = -2(with 1 spins up and 3 spin down) in 2×2 lattice, we may have

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Revise the loss function (cont.)

Those four configurations may not be entairly included due to our finite sample size.
e.g. we may miss the last configuration

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} \cancel{0} & \cancel{0} \\ \cancel{0} & \cancel{1} \end{pmatrix}$$

All of those three configurations are mapped to the same hidden variable, and the final output should be

$$\begin{pmatrix} 0.33 & 0.33 \\ 0.33 & 0 \end{pmatrix}$$

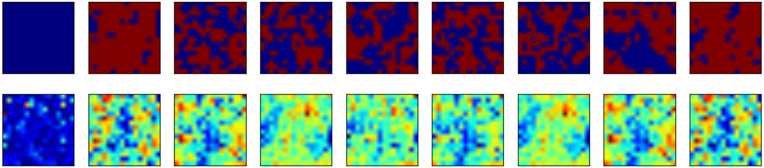
when we minimize the loss function above. This is exactly the kind of configurations I got in my 16×16 lattice.

Revise the loss function (cont.)

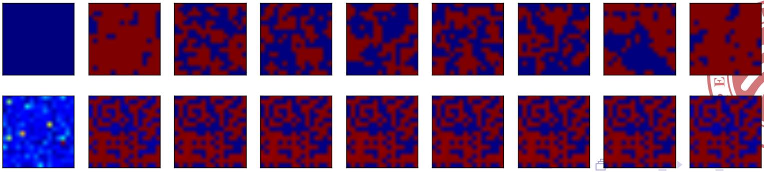
- I add a constrain term into the loss function

$$Loss = L_2 norm + \alpha[-\sum_{i}^{N}(\sigma_i - \frac{1}{2})^2] \quad (+KLD)$$

This turns the result from



to



Revise the loss function (cont.)

We may illustrate such result using the 2×2 lattice.
at this time, when M=0 and -2, our samples are

$$\begin{pmatrix}1 & 0\\1 & 0\end{pmatrix} \quad \begin{pmatrix}1 & 1\\0 & 0\end{pmatrix} \quad \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$$

$$\begin{pmatrix}1 & 0\\0 & 0\end{pmatrix} \quad \begin{pmatrix}1 & 0\\0 & 0\end{pmatrix} \quad \begin{pmatrix}0 & 0\\0 & 1\end{pmatrix}$$

to minimize the new loss function, the result should be

$$\begin{pmatrix}1 & 0\\0 & 0\end{pmatrix}$$

for both M. Then here comes the mode collapse.

Revise the loss function (cont.)

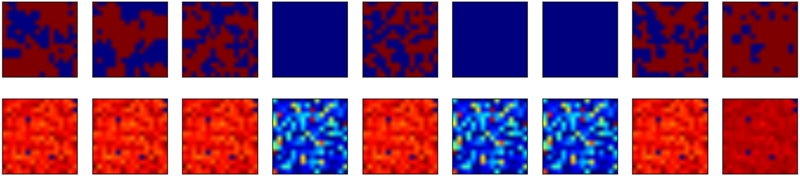
So, the mode collapse effect comes from the L2 penalty term. We can replace this term with magnetic moment or other human knowledge, but using human knowledge is not very elegant.
Training networkB using knowledge from networkA is another choice.

1. Use a self-supervised CriticNet[1], which can be a AE/VAE that has the L2 loss function, to learn about the potential law of the input data.
2. Then, use another AE/VAE, called ActorNet[1], with its loss function difined by CriticNet's encoder.
3. This idea can be extended to other situations where we do not understand the underlying law or with more hidden variables to improve the precision.

[1] I stole these nouns from the terminology related to reinforcement learning

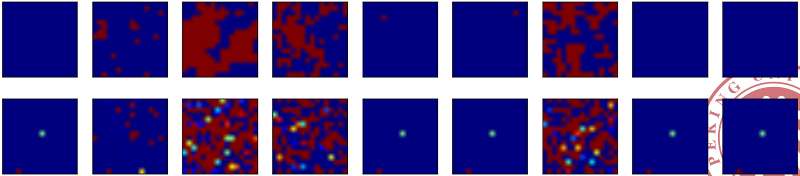
Result

- With AE as the CriticNet, AE as ActorNet:



Which shows AE can not map new configurations to the hidden variable, this mapping is not smooth.

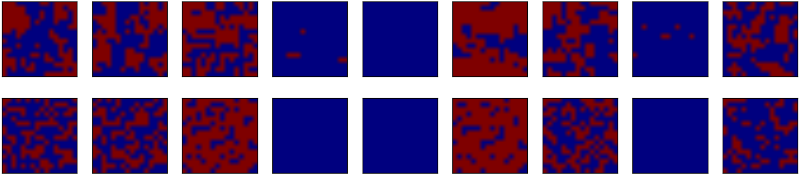
- With VAE as the CriticNet, AE as ActorNet:

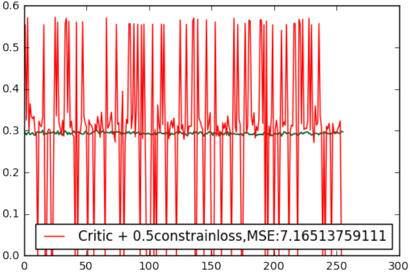


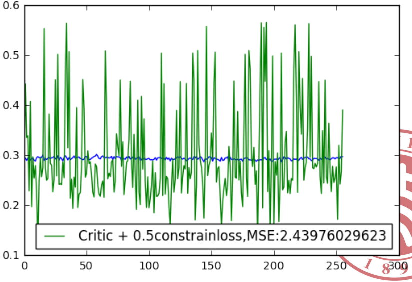
VAE is good at understanding a new configuration.

VAE as Critic and MLP as Actor

Instead of using AE or VAE, I tried to use MLP for the ActorNet

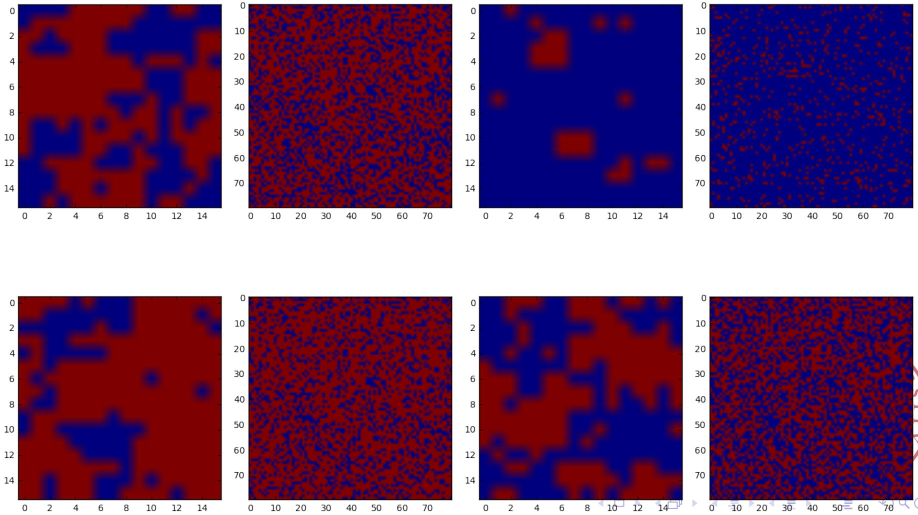






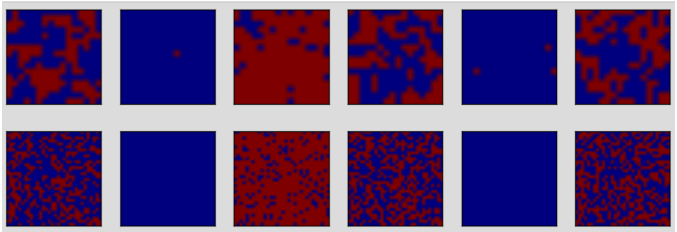
Scale expansion (Mode collapse)

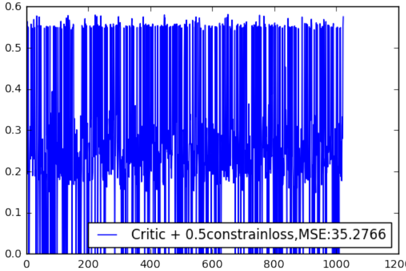
Use AE for the ActorNet to generate 4*4*N spin configurations.

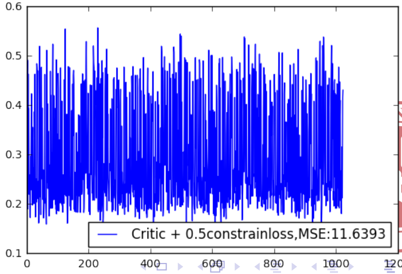


Scale expansion (5% collapse)

Use MLP for the ActorNet to generate 2*2*N spin configurations.







Summary and Problems

- Mode collapse still exists, but can be partially solved by using more generators(more ActorNets), for each of those generators has fixed finite weights.
- Finished generating 25*20,000 samples of N=32*32 Ising model spin configurations in 1133s (can be accelerated by GPUs). Such generators can save 99% of time than MCMC. But sampling bias is the cost.
- When N=256, the configurations are sampled from the whole $\mathscr{F}_2^{256}$ space. If N=25,600, the configurations will be sampled from the whole $\mathscr{F}_2^{25600}$ space. Neural networks may help to generate more datas on the same scale. But the representation ability is still insufficient for much higher dimension.