Deep Learning in Physics
深度学习在物理学中的应用

Hao Sun[†][*]

[†]School of Physics@Peking University

[*]Yuanpei College@Peking University

May 23, 2018

Contents

Contents

A brief Introduction of Ising Model

- Ising Model is one of the most widely studied models in statistical physics. It was first proposed by Lenz in 1920 and calculated by Ising.
- It was originally used to study the behavior of magnetic objects in the magnetic field. It is the simplest theoretical model of ferromagnets.
- In the two-dimensional Ising Model, considering the $L \times L$ spin particles at the lattice point position, the spin of each particle can only be up or down.Ising Model only consider the interaction between the nearest neighbor spin. Its Hamiltonian is

$$H = -J \sum_{<i,j>} s_i s_j - \mu B \sum_i s_i \tag{1}$$

Contents

Metropolis Algorithm

Algorithm 1 Simulate L*L Ising Model Spin Configuration Use MCMC

Require: simulation scale L, repeat sample time M
Ensure: spin configuration

```
for each t in 0,1,2,...,M do
    for each i in 1,2,3,...,L*L do
        flip a spin randomly, calculate ΔE = E_1 − E_0
        if ΔE < 0 then
            accept the flip
        else
            if exp(−ΔE/kT) < random(0,1) then
                reject the flip
            else
                accept the flip
            end if
        end if
    end for
end for
```

HMC Algorithm

Algorithm 2 Simulate L*L Ising Model Spin Configuration Use Synchronous HMC

Require: simulation scale L, repeat sample time M
Ensure: spin configuration

```
for each t in 0,1,2,...,M do
    for each i in 1,2,3,...,L*L do
        S_i(t+1) = sign(∑_{j=1}^{4} S_i S_j − T · ln(1/random(0,1))) · S_i(t)
    end for
end for
```

$$S_i(t+1) = \mathrm{sign}(\textstyle\sum_{j=1}^{4} S_i S_j - T \cdot \ln \frac{1}{random(0,1)}) \cdot S_i(t)$$

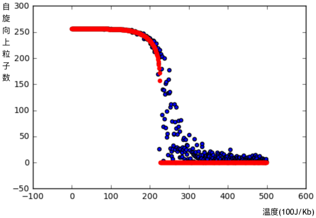
Algorithm 3 Simulate L*L Ising Model Spin Configuration Use Asynchronous HMC

Require: simulation scale L, repeat sample time M
Ensure: spin configuration

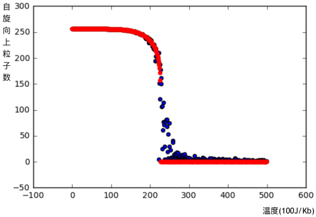
```
for each t in 0,1,2,...,M do
    for all i in L*L configuration do
        S_i(t+1) = sign(∑_{j=1}^{4} S_i S_j − T · ln(1/random(0,1))) · S_i(t)
    end for
end for
```

$$S_i(t+1) = \mathrm{sign}(\textstyle\sum_{j=1}^{4} S_i S_j - T \cdot \ln \frac{1}{random(0,1)}) \cdot S_i(t)$$

Comparison Between MCMC and HMC



(a) MCMC, 1e3 steps, MSE=827, time=4191s



(b) HMC-P2, 1e3 steps, MSE=461, time=2017s

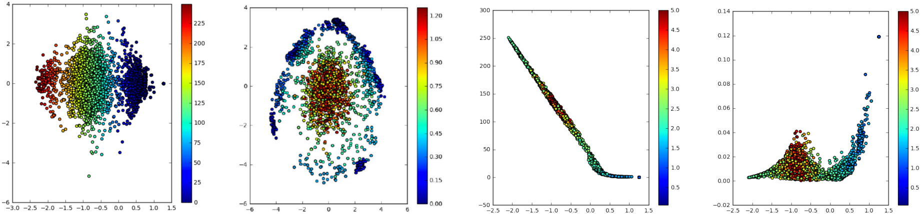
Figure 1: Result Comparison between MCMC and HMC-P2

	Iteration Steps	Time ⁄ s	MSE
(1)MCMC	1000	297	272
(2)HMC	4000	311	133
(3)HMC	1000	77	197
(4)HMC-P2	2500	284	127
(5)HMC-P2	1000	111	143
(6)HMC-P2	**10**	**1.13**	294

Table 1: Comparison of MCMC, HMC and HMC-P2

Contents

Finding Order Parameters Using Unsupervised Learning



(a) Hidden variables of 2D-Ising model (b) Hidden variables of 2D-XY model (c) Mean of Hidden variables of Ising Model (d) Var. of Hidden variables of Ising Model

Figure 2: Unsupervised learning result

Result

Some laws of statistical physics can be discovered by generative models without human knowledge. Such as order parameter of Ising model.

Contents

Generate Ising Model Spin Configurations Using VAE

The loss fucntion of VAE is

$$L_{\text{VAE}} = \| X_{\text{in}} - X_{\text{out}} \|^2 + D_{\text{KL}}(P(V), \mathcal{N}(0, I)) \tag{2}$$

By minimizing the loss function above, there exists obvious mode collapse in the spin configurations generated:

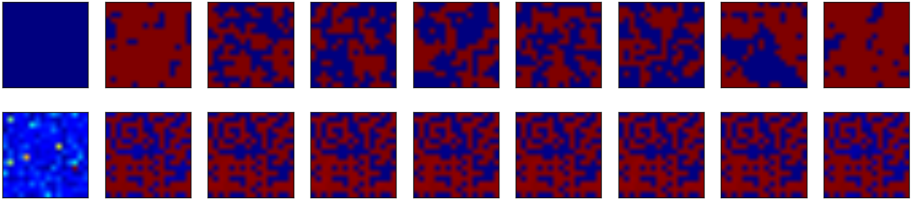


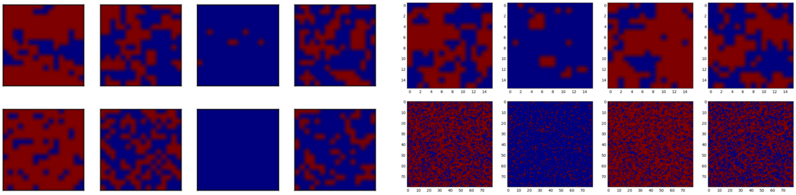
Figure 3: Spin configuration generated by VAE

Generate Ising Model Spin Configurations Using Actor-Critic Net

Using the Actor-Critic Net proposed in my thesis, the loss function was revised:

$$L_{\text{Actor}} = \| \, CriticNet(X_{\text{in}}) - \overline{CriticNet(\mathrm{Scale}(X_{\text{out}}))} \, \|^2 \tag{3}$$

which is more suitble for such generation tasks, and the result for the same scale become:



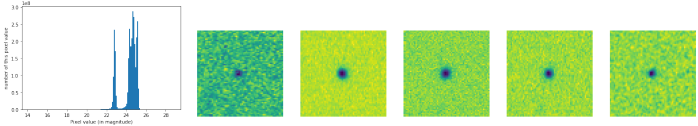
(a) AC-net generation result at the same scale (b) AC-net generation result at large scale

Figure 4: Result of Actor-Critic net

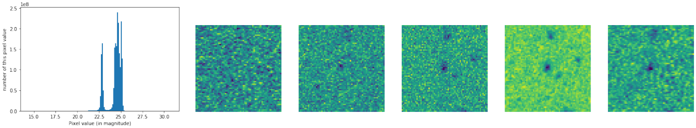
Contents

An Introduction to the Data Set

The dataset we choose to use in this work is the SDSS dataset, including a labeled sub-dataset $X_{labeled}$ and an unlabeled sub-dataset $X_{unlabeled}$.



(a) Data set A, including 140,000 imgs, size $64 \times 64 \times 5$, with label



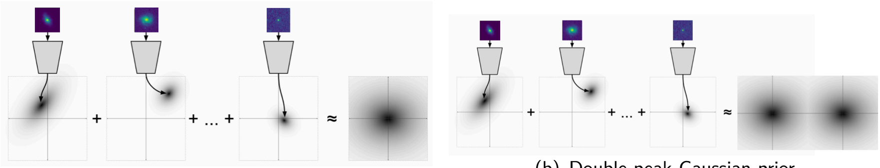
(b) Data set B, including 100,000 imgs, size $64 \times 64 \times 5$, without label

Figure 5: Comparison of the two data set

Contents

Revise the KL-Divergence term in VAE

With the Gaussian prior, we can sample from the Gaussian to generate a consistent image that at least looks like one of the input images in the training set. This $\mathcal{N}(0,1)$ prior works well in generating new images, but is not conducive to unsupervised classification. To separate different classes into two



(a) Normal distribution prior

(b) Double peak Gaussian prior

Figure 6: Hidden space distribution with different prior

And in my VAE structure I set the prior

$$Q(V) \sim \frac{1}{2}N(-m, s^2) + \frac{1}{2}N(m, s^2) \tag{4}$$

An Hierarchical Design of the Network

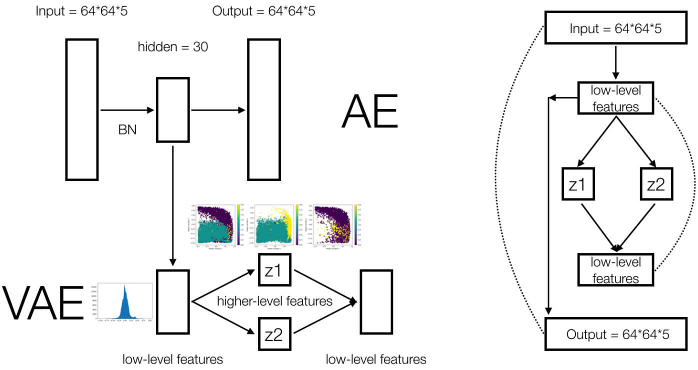
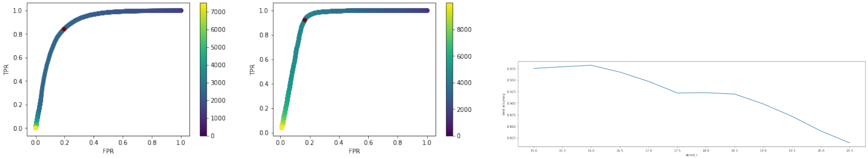


Figure 7: Structure of the network

Classification Result

Table 2: AUC comparison of different method

Method	mean	Highest	Lowest	Std	repeat time
VAE + Manifold Learning	0.76	0.89	0.56	0.1	10
AE+VAE with revised KLD	**0.90**	**0.91**	**0.90**	**0.002**	100



(a) Vae+Manifold Learning: AUC=0.89

(b) AE+VAE with revised KLD:AUC=0.91

(c) Accuracy according to Megnitude

Figure 8: Classification Result

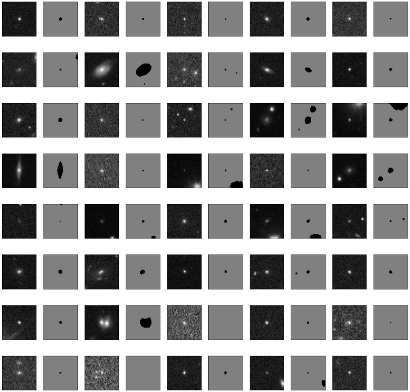
Result

The average AUC is much better than traditional unsupervised method, with much smaller standard deviation. The accuracy 91% is quite close to TSOA supervised result 97%.

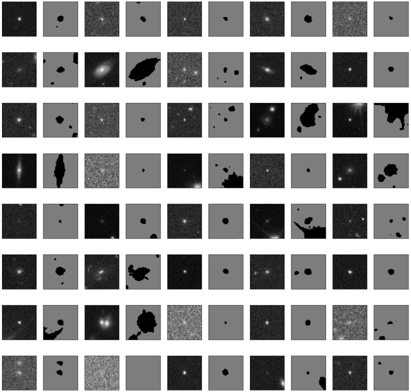
Contents

Unsupervised star-galaxy segmentation

Use Auto-Encoders and Hypercolumns. Tried to use residual connection as comparison



(a) AE + Hypercolumns



(b) AE + residual connection + Hypercolumns

Figure 9: Segmentation Result

Combination of the Segmentation and Classification Result

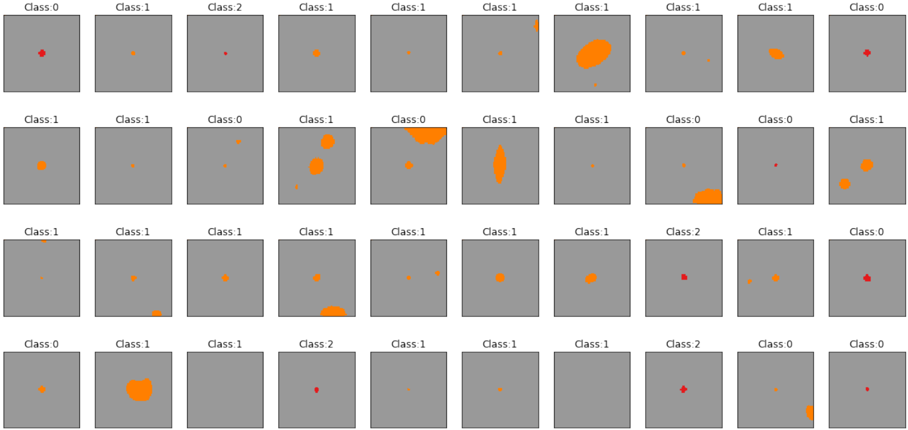


Figure 10: Segmentation with Classification

Contents

Introduction to PROMISE12 Chanllenge

MICCAI Prostate MR Image Segmentation challenge datase is an active prostrate segmentation test

- Training Set: 50 3D MR images with different size and corresponding segmentation labels
- Test Set: 30 3D MR images with different size

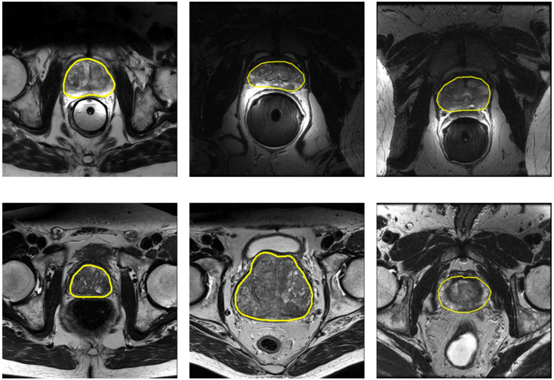
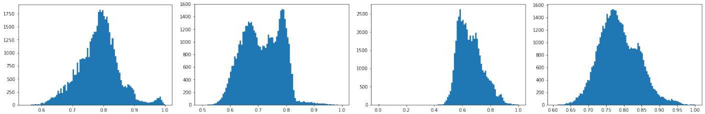


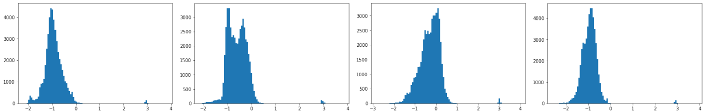
Figure 11: Sample images

Contents

Normalization Method and Loss Function



(a) Histgram of pixel values normalized to 0-1



(b) Histgram of pixel values normalized by CNN

Figure 12: Histgram of pixel values with different nomalization methods

Using Dice Coefficient(DSC) instead of MSE/Cross Entropy as loss function

$$D = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} \tag{5}$$

Contents

Neural Net Structure

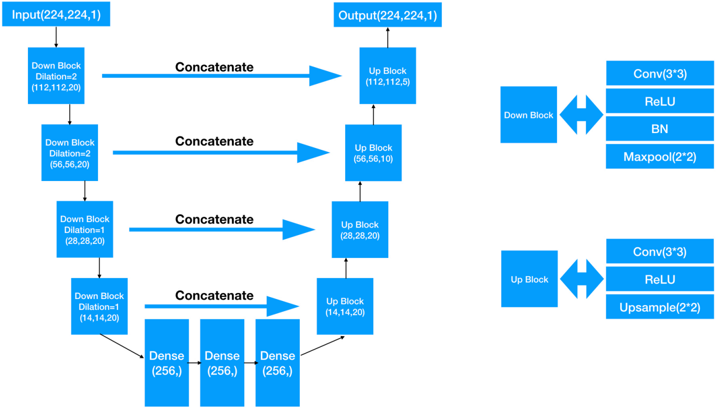


Figure 13: The neural net structure designed for the segmentation task (with 4 layers)

Segmentation Result

layers	down kernels	up kernels	dilation	residual	DSC	DSC(BN)
4	20,20,20,20	20,10,5,1	2,2,1,1	conv1-4	0.755	0.7899
4	20,20,20,20	20,20,10,5,1	2,2,1,1	conv1-4	**0.7789**	0.8257
5	20,20,20,20,20	20,10,10,10,5,1	2,2,1,1	conv1-4	0.7512	**0.8455**
5	20,20,20,20,20	20,10,10,10,5,1	2,2,1,1	conv1-5	0.762	0.8383
5	20,20,20,20,20	20,10,10,5,5,1	2,2,1,1	conv1-5	0.729	0.7939
5	20,20,20,20,20	20,10,10,10,5,1	2,2,2,1	conv1-5	0.755	0.7994
5	40,40,20,20,20	20,20,20,20,10,1	2,2,1,1	conv1-4	0.762	0.8318
5	20,20,20,20,20	20,10,10,10,10,1	2,2,1,1	conv1-5	0.7601	0.8197

Table 3: Result with different hyperparameter settings

Result

When facing the challenge of small training set as well as the problem of multi-size of datas, my thesis proposed the method of normalization using CNN with BN layers. With this method, the DSC of normal neural net structure can be rased by 0.04-0.1. And the training process get much more stable.