Relation of machine learning and renormalization group in Ising model

Shotaro Shiba (KEK)

Collaborators: S. Iso and S. Yokoo (KEK, Sokendai)

Reference: arXiv: 1801.07172 [hep-th]

Jan. 23, 2018 @ Osaka Univ.

What is machine learning ?

- Fundamental motivation is "to clarify the mechanism of consciousness." There are two directions of research.

1. We may understand it from reactions of human bodies or matters. --- Structuralism, Experimentalism
2. If we can reproduce actions of consciousness, we may understand its mechanism. --- Functionism

- In 1970's, the former research get stuck, then the latter research get active.
- Research to reproduce actions of consciousness on computers = Research on artificial intelligence (AI)

- For designing an AI, there are (again) two directions.

1. By teaching a machine on various knowledge and rules, we may design an AI which can judge things like humans.
2. By emulating a structure of human brain, we may design an AI which can learn and judge various information. (Machine learning)

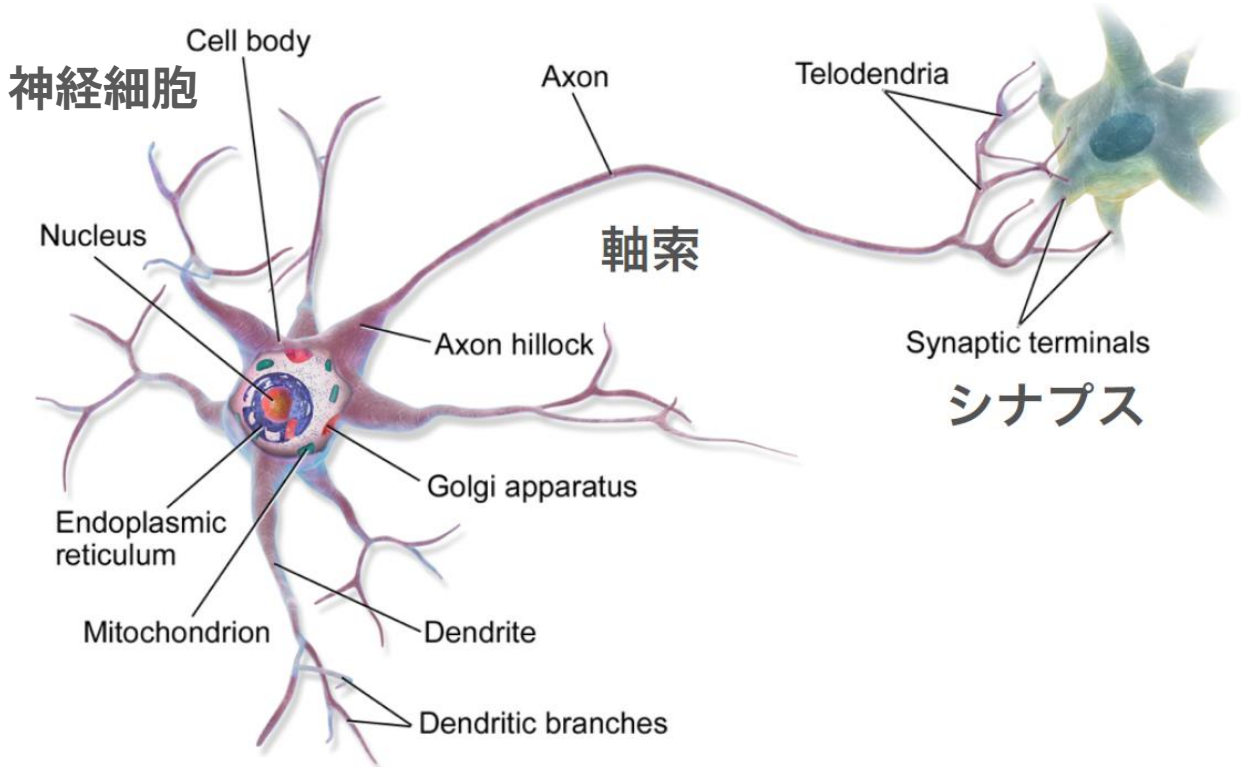
It finds the rules!

- Both researches continue to be done, but recently the latter has been greatly developed.

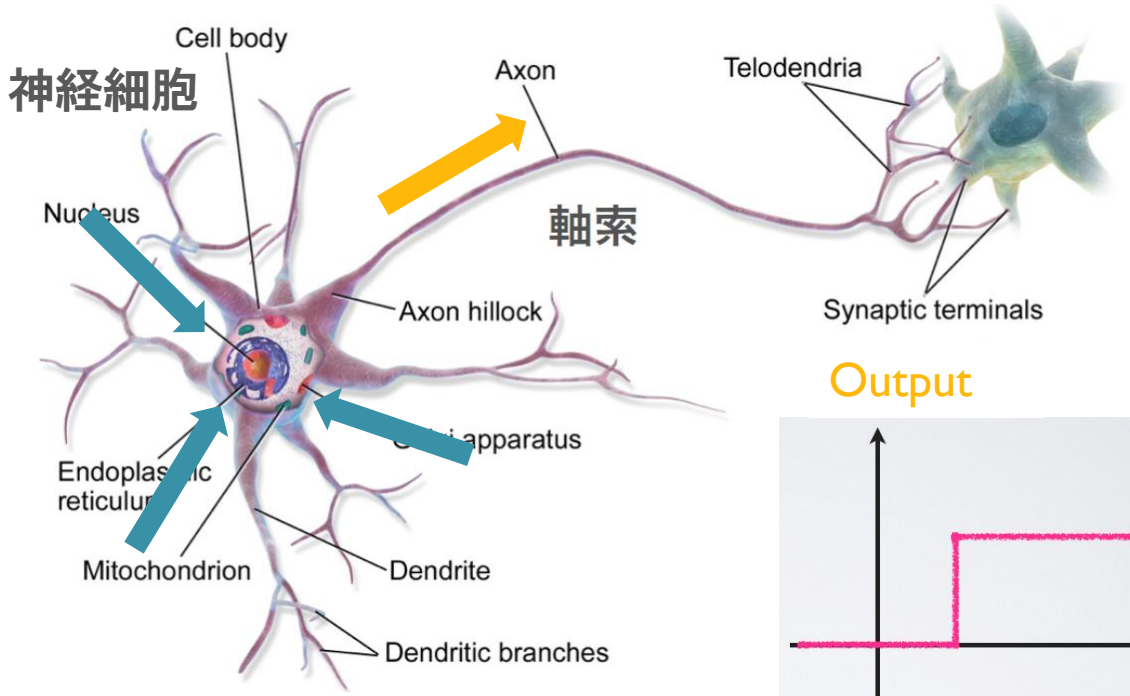
- Artificial intelligence is "incomplete" yet, but it has begun to give some influence on our daily life.
- My motivation is… "I want to contribute something from the viewpoint of particle physics."

Structure of human brain

- Human brain has about 100 billion neurons（神経細胞）and they are connected via axons（軸索）.

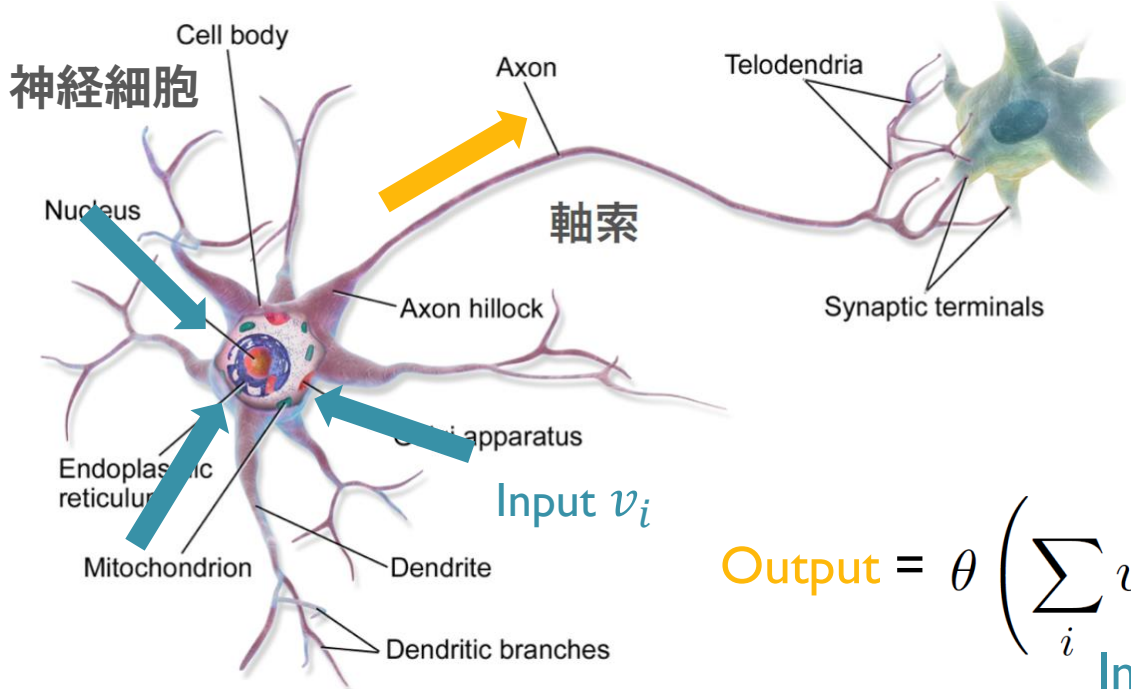


- A neuron receives electric signals sent from other neurons through axons.
- If a sum of input signal exceeds a threshold (certain value), the neuron fires and sends a signal to other connected neurons by axons.



- We everyday repeat trials and errors. After each of such experiences, our neurons renew a way to exchange the signals so that we can grow to judge various things more properly.

→ This is nothing but "learning."



Input v_i

Adjusted by "learning"

weights w_i

bias (threshold) b

$$\text{Output} = \theta\left(\sum_i v_i w_i + b\right)$$

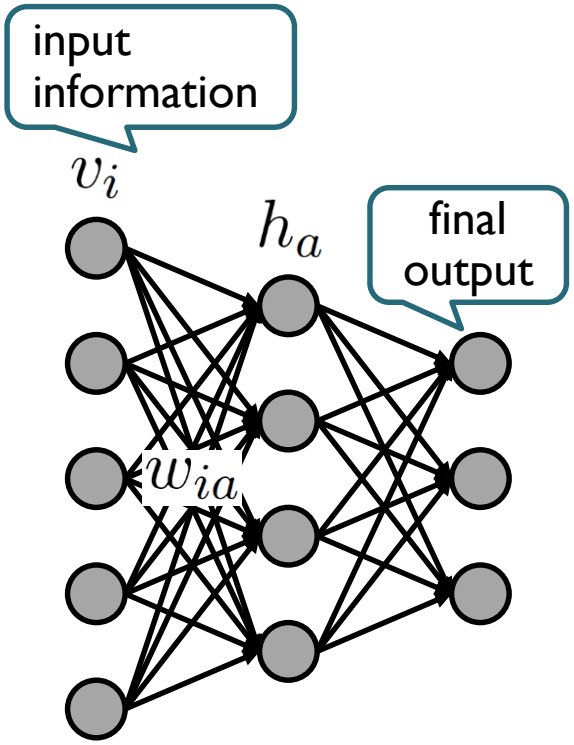
Input v_i

Algorithm of machine learning

- By emulating such a structure of brain, we design an algorithm of machine learning.
- In particular, we reproduce a network of neurons exchanging signals. For example…

$$h_a = f\left(\sum_i v_i w_{ia} + b_a\right)$$

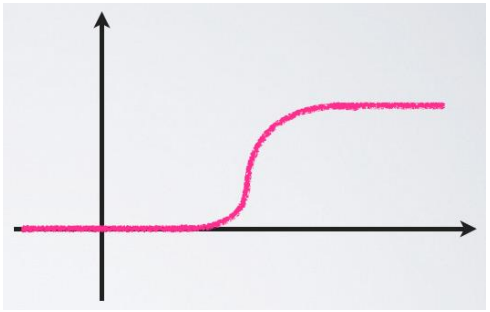
nonlinear function (activation function)

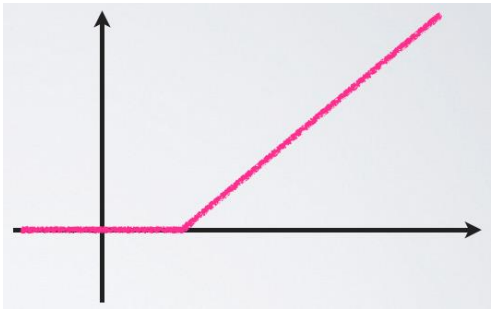


- We adjust weights w_{ia} and bias b_a so that the final output approaches (a set of) desired values for us.

"training"

- As an activation function (nonlinear function), because of analyticity, we don't often use the step function but sigmoid function (left) or ReLU (right).





$$f(x) = \frac{1}{1+e^{-x}}, \qquad f(x) = \begin{cases} x & \text{for } x \geq 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

- For the final output, the softmax function is often used, since we can interpret it as probability.

$$g(x_i) = \frac{\exp x_i}{\sum_j \exp x_j}$$

- In order to adjust values of weights w_{ia} and bias b_a…

- We choose the loss function which evaluates difference between output at present and desired output.
  Square sum or relative entropy is often chosen.

  For probability distributions (later)

- Then we calculate values of weights and bias such that the loss function takes the minimum value (zero, ideally).
  However, it is almost impossible to calculate it analytically.
  In general the weights and bias have a lot of dofs, then we need to solve nonlinear equations with many parameters.

- Instead, using numerical calculations, we make an algorithm to find the minimum (practically a local minimum) by iterative approximation.

"training"

Method of back propagation

- The output is given as a composite function of nonlinear functions.

$$\begin{aligned} v_L &= W_L v_{L-1} =: u_L \\ &= W_L\, f(W_{L-1} v_{L-2}) =: W_L\, f(u_{L-1}) \\ &= \cdots \\ &= W_L\, f(W_{L-1}\, f(W_{L-2} \cdots f(W_2\, f(W_1 v_0)) \cdots)) \end{aligned}$$

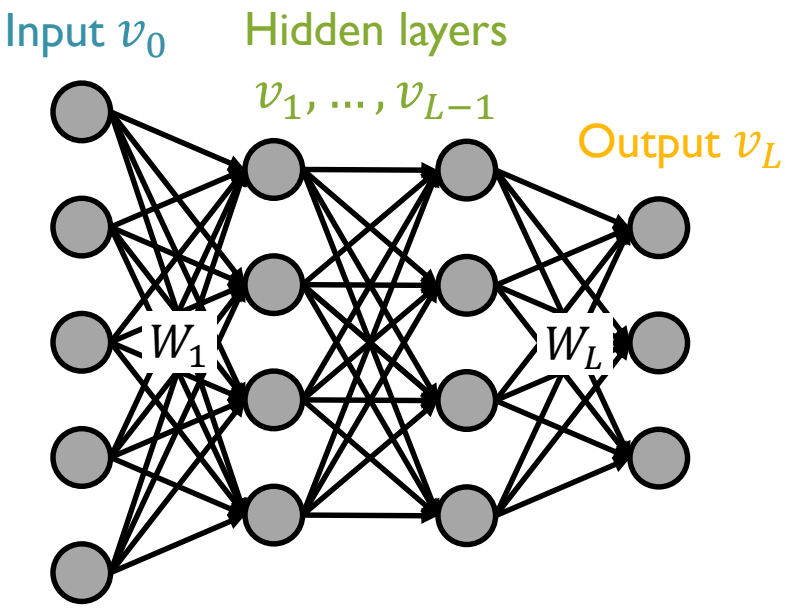
We neglect bias, for simplicity

- We choose the loss function to evaluate difference of the output v_L and desired one y.

$$E = \frac{1}{2} \sum_n \left(v_L^{(n)} - y^{(n)} \right)^2$$

Weights should be renewed by iterative approx.

$$W_\ell \to W_\ell - \eta \frac{\partial E}{\partial W_\ell}$$



- To obtain weights such that the loss function takes a local minimum, we need to calculate derivatives of a composite function.

$$\frac{\partial E}{\partial W_\ell} = \frac{\partial E}{\partial u_\ell}\frac{\partial u_\ell}{\partial W_\ell} = \frac{\partial E}{\partial u_\ell} v_{l-1}$$

- The back propagation is a technique to calculate them easily. We also consider derivatives w.r.t. $u_l = W_l v_{l-1}$:

$$\frac{\partial E}{\partial u_\ell} = \frac{\partial E}{\partial u_{\ell+1}}\frac{\partial u_{\ell+1}}{\partial u_\ell} = \frac{\partial E}{\partial u_{\ell+1}} W_{\ell+1} f'(u_\ell)$$

- This is a recurrence formula for u_l. Then we can calculate the derivatives in order of $l = L, L-1, \ldots, 2, 1$.

"back" propagation

$$\frac{\partial E}{\partial u_L} = \sum_n \left(v_L^{(n)} - y^{(n)} \right)$$

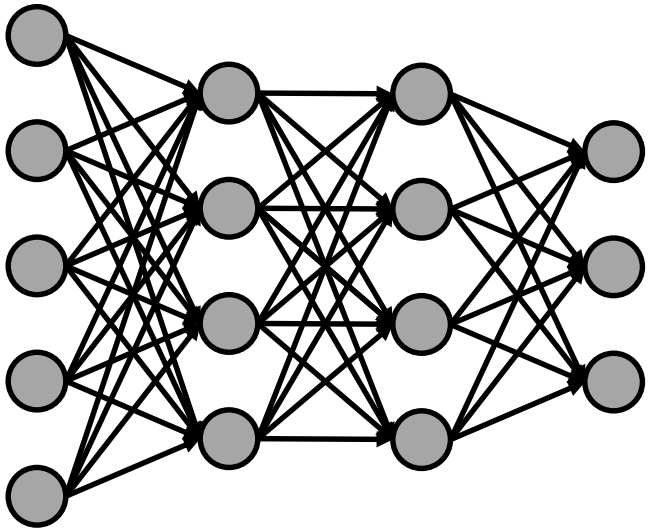
But it goes for some reason!

- However we use iterative approx., so no one can assure that the loss function approaches the (global) minimum…

Google's cat (2012)

- Using such algorithms, we can get interesting results.
  For example, …

~10 million still images clipped from YouTube movies are input.



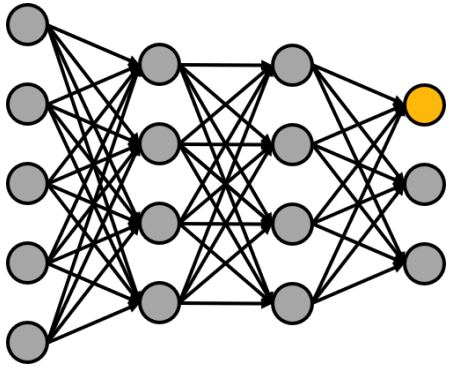
Humans didn't teach anything!

It learns so as to output images as similar as possible to the inputs. (autoencoder)

A network of neurons (neural network, NN) with more than 10,000 layers

- What kind of images does each neuron react to?

- If we make an input image such that only specific neurons react, in the image appears a human face or a cat's face. This means there are neurons reacting to humans or cats.
- There are also neurons which react to simpler figures, such as a line, an edge or a triangle. In general, neurons in deeper layer (close to output) reacts to more complicated things.
- This seems to reproduce a human process of grasping "characteristics" and recognizing "concepts."



- What does grasping "characteristics" mean?

- Of all information in an image, we extract an important part as its features and drop the other parts.

- It seems similar to the coarse graining.
  That is, it may be related to the renormalization group (RG).
  Going along the RG flow, relevant parameters are emphasized while irrelevant parameters are dropped.

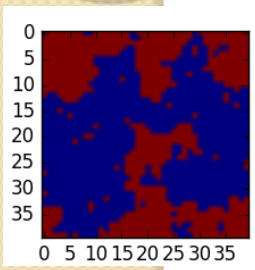
- Perhaps we can discuss something from the viewpoint of particle physics…!

[Mehta-Schwab, '14]
[Lin-Tegmark-Rolnick, '16]
[Sato, '16]
[Aoki-Kobayashi, '16]
[Koch-Janusz, Ringel, ' 17]

Our experiment and results

Our experiment (1)

[Iso-SS-Yokoo, '18]



1. Using numerical calculations, we generate samples of spin configurations of 2d Ising model at temperatures T=0, 0.25, …, 6. (These configurations are described as black-and-white images.)
2. We design a NN which outputs images as similar as possible to inputs when the configurations are input. (autoencoder, unsupervised learning)
3. We input again the output configurations to the NN. Doing this iteratively, we obtain the flow of reconstructed configurations.

Generating spin configurations

- We consider spin configurations in 2d Ising model with the size of 10x10, imposing a periodic boundary condition.
- Hamiltonian for ferromagnetic (J>0) is used, where neighbor electrons tend to have same spins at low temp.

$$\sigma_{i,j} = \pm 1$$

$$H = -J \sum_{i,j=0}^{L-1} \sigma_{i,j} \left(\sigma_{i+1,j} + \sigma_{i-1,j} + \sigma_{i,j+1} + \sigma_{i,j-1} \right)$$

- We generate spin configurations by Monte Carlo simulation. Beginning with a random config, we flip the spins $\sigma_{i,j} \to -\sigma_{i,j}$ with probability $p_{i,j}$.

$$dE_{i,j} = 2J\sigma_{i,j} \left(\sigma_{i+1,j} + \sigma_{i-1,j} + \sigma_{i,j+1} + \sigma_{i,j-1} \right)$$

$$p_{i,j} = \begin{cases} 1 & (\text{when } dE_{i,j} < 0) \\ e^{-dE_{i,j}/k_B T} & (\text{when } dE_{i,j} > 0) \end{cases}$$

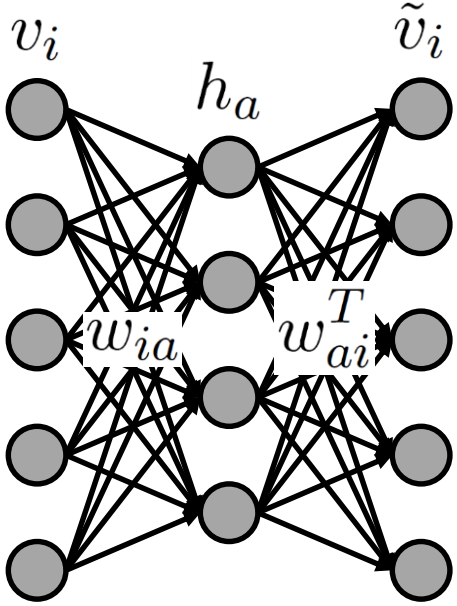
For various temperature T

Autoencoder (unsupervised learning)

- An autoencoder, which plays important roles in "Google's cat," is believed to extract "features" of input images.

What is the definition of features?

- It can be regarded as a NN which compresses the images and then reconstructs them.
- Here we design a NN which outputs the same images as inputs with the same probability (as inputs).
- This type of autoencoder is called Restricted Boltzmann Machine (RBM).



- The probability of outputting an image is defined, using the energy function

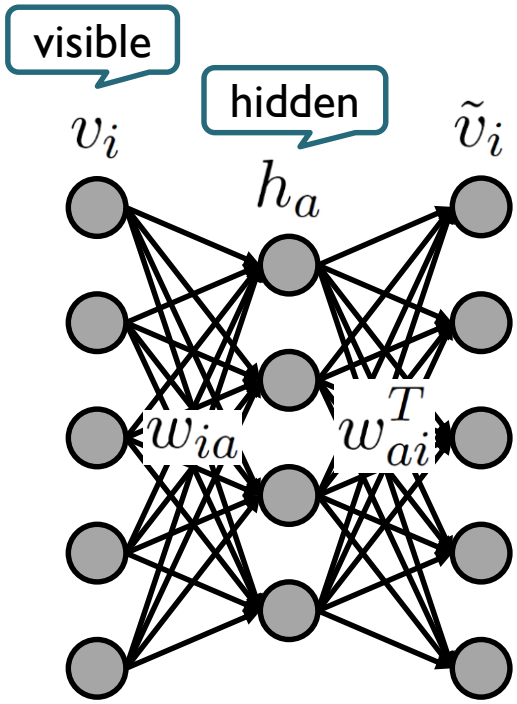
$$E(\{v_i\},\{h_a\}) = \sum_{i,a} v_i w_{ia} h_a + \sum_a b_a h_a + \sum_i c_i v_i$$

weights w_{ia}, bias b_a, c_i

by Boltzmann distribution

$$p(\{h_a\}) = \sum_{\{v_i\}} \frac{e^{-E(\{v_i\},\{h_a\})}}{\mathcal{Z}}$$

$$\tilde{p}(\{\tilde{v}_i\}) = \sum_{\{h_a\}} \frac{e^{-E(\{\tilde{v}_i\},\{h_a\})}}{\mathcal{Z}}$$



- We train the RBM (weights and bias) so that the relative entropy approaches the local minimum.

$$\sum_{\{v_i\}} q(\{v_i\}) \log \frac{q(\{v_i\})}{\tilde{p}(\{v_i\})}$$

25000 configs are used for training

- The relative entropy is also called KL divergence.

$$\sum_{\{v_i\}} q(\{v_i\}) \log \frac{q(\{v_i\})}{\tilde{p}(\{v_i\})}$$

prob of an input image = v_i / prob of an output image = v_i

- In our experiment, the input images are white-and-black images of spin configurations in 2d Ising model: $v_i = \pm 1$.
- The outputs show the expectation values of spins.

$$\langle h_a \rangle = \tanh\left(\sum_i v_i w_{ia} + b_a\right)$$

$$\langle \tilde{v}_i \rangle = \tanh\left(\sum_a h_a w_{ai}^T + c_i\right)$$

- The final output (reconstructed) images have spins $\tilde{v}_i = \pm 1$ by replacing an EV $\langle \tilde{v}_i \rangle$ with a probability $(1 \pm \langle \tilde{v}_i \rangle)/2$.

KL divergence becomes small but not zero!

- The probability distribution of input images $q(\{v_i\})$ and that of output images $\tilde{p}(\{v_i\})$ are similar but slightly different.
- If we input again the output images to the RBM, we obtain another probability distribution $\tilde{\tilde{p}}(\{v_i\})$ of reconstructed images.
- Then, doing this procedure iteratively, we get the flow of probability distributions: $q(\{v_i\}) \to \tilde{p}(\{v_i\}) \to \tilde{\tilde{p}}(\{v_i\}) \to \dots$

25000 configs (1000 for each T) which are not used for training

- A naïve expectation:

  The RBM flow may correspond to the RG flow, especially when the size of hidden layer n_h is smaller than that of visible layer n_v (since the RBM *compresses* images).

Our experiment (2)

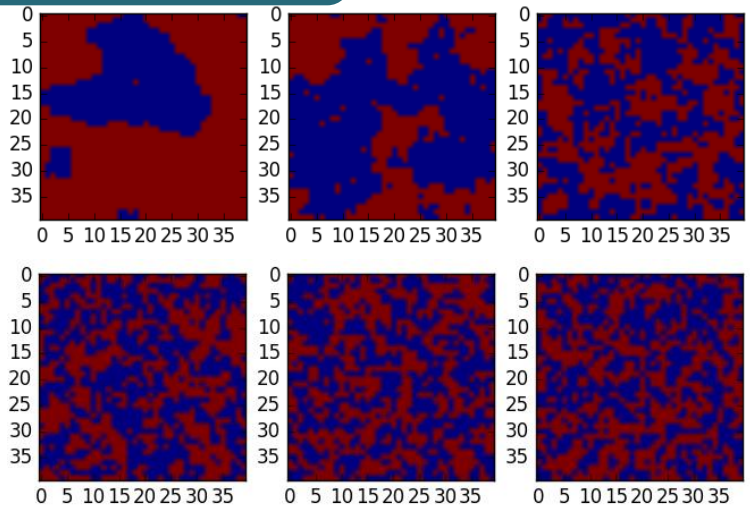
[Iso-SS-Yokoo, '18]

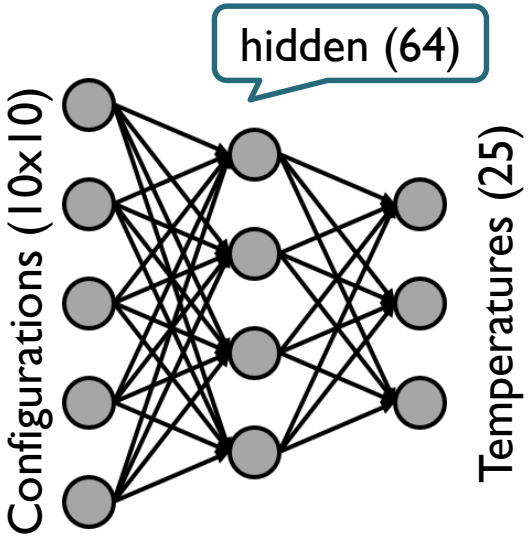
- We want to check whether the RBM flow is related to the RG flow.
- Let us translate the flow of probability into a flow of temperature, which makes this discussion easier.
- To do this, we design another NN which outputs correct temperature of an input configuration. (supervised learning)
- We input all of the original and reconstructed configurations (by the RBM) into this NN to measure their temperature.
- Then the flow of probability distributions can be translated into a flow of temperature distributions $T(\{v_i\}) \to \tilde{T}(\{v_i\}) \to \tilde{\tilde{T}}(\{v_i\}) \to \dots$

NN for temperature measurement

- We design a NN which outputs the probability that an input image is a configuration at each of 25 temperatures T=0, 0.25, 0.5, …, 6. (The softmax function is used for the final output.)
- We train the NN so that the probability of correct temp is large enough. (The relative entropy is used as the loss function.)

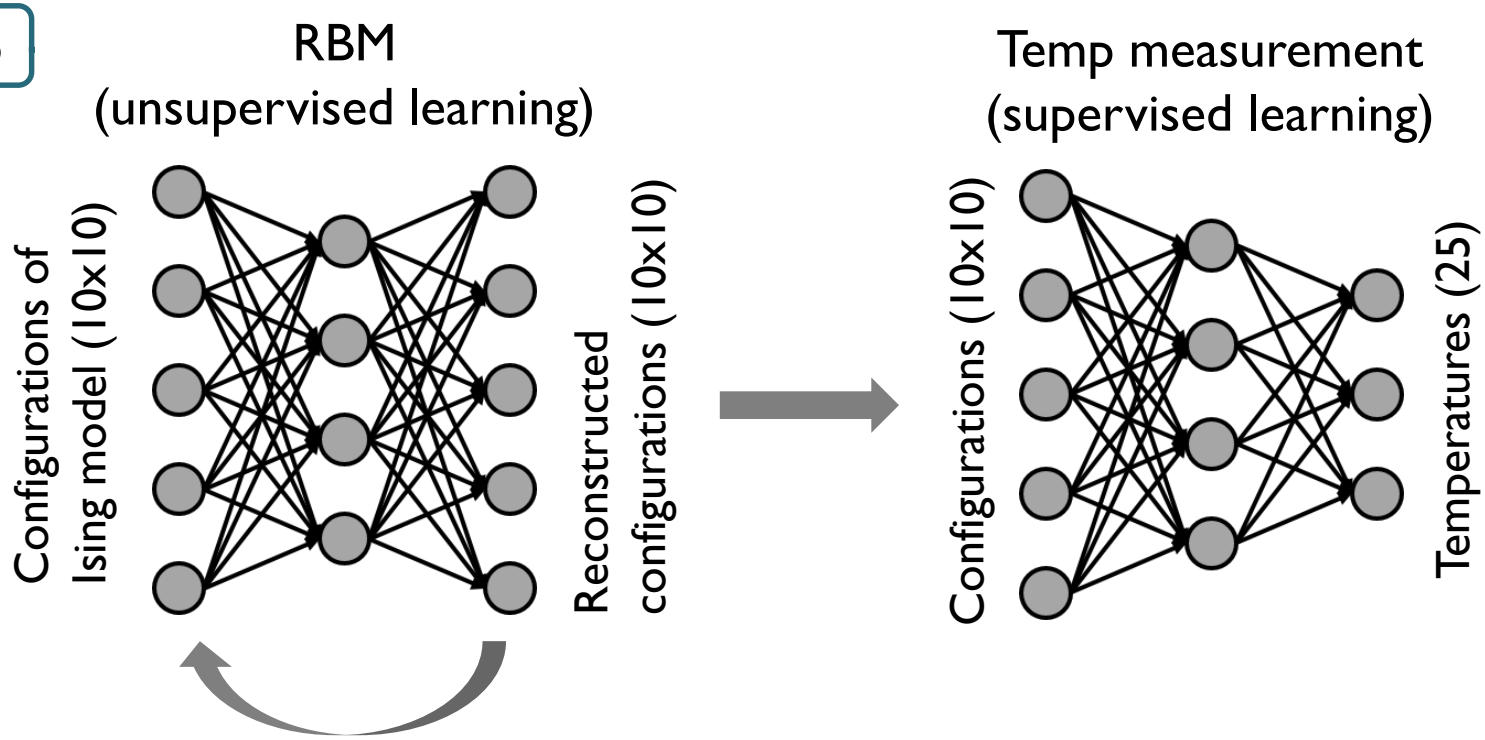
We use 25000 configs for training, and other 25000 configs for test.





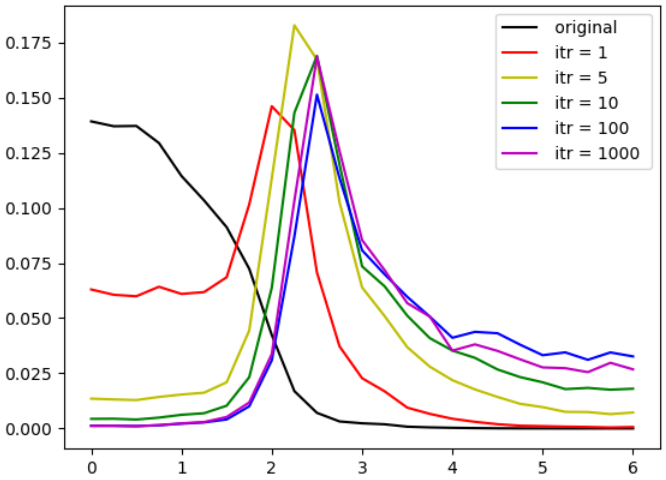
- If the RBM corresponds to the renormalization (as expected), the RBM flow of temperature corresponds to the RG flow in 2d Ising model.
- If so, the RBM flow of temperature (of reconstructed configs) would go away from the critical temperature T=2.27.

Our setup

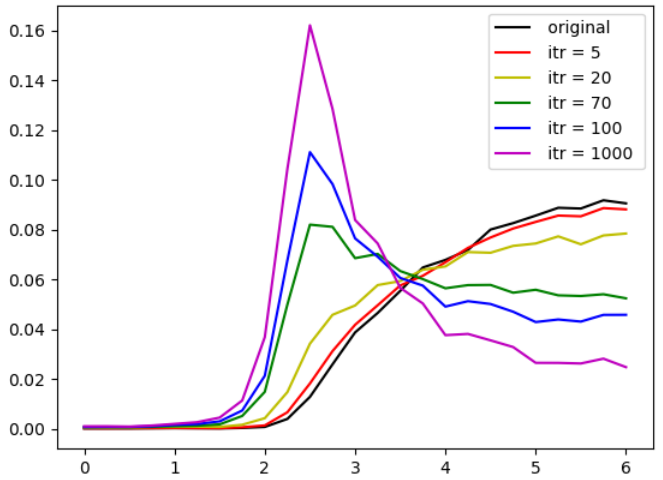


Result: RBM is different from RG!

- When we use an RBM with $n_h \leq n_v = 100$ neurons, the flow of temperature approaches the critical point T=2.27.
- If the RBM has $n_h = 81$ neurons in hidden layer and we input the configurations at T=0 and T=6, … (movies)

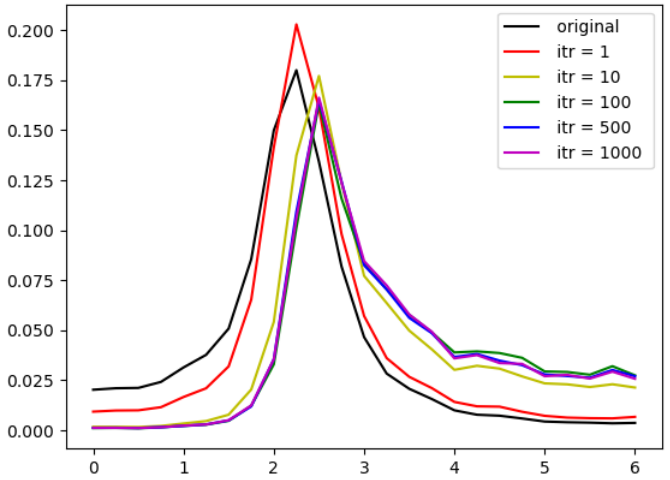


Starting from T=0 configs



Starting from T=6 configs

- We find that the RBM flow is in the opposite direction to the conventional RG flow: This seems an interesting result!
- A lot of questions come up to our mind… For example: We didn't teach the RBM anything about the phase transition, but the RBM flow behaves as if it knows about the critical temperature. Why does it happen?
- In addition, what is a feature which the RBM extract from the input configurations?

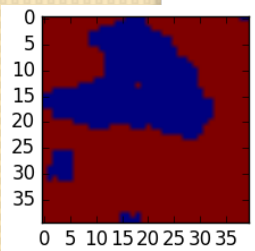


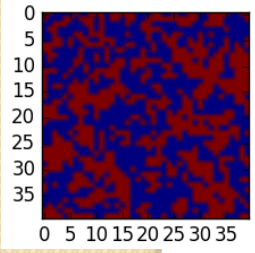
Starting from T=2.25 configs

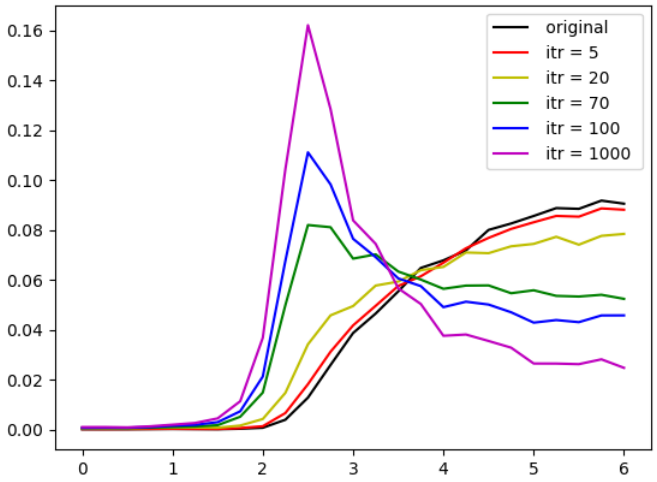
Analysis of our results

Feature for RBM: scale invariance?

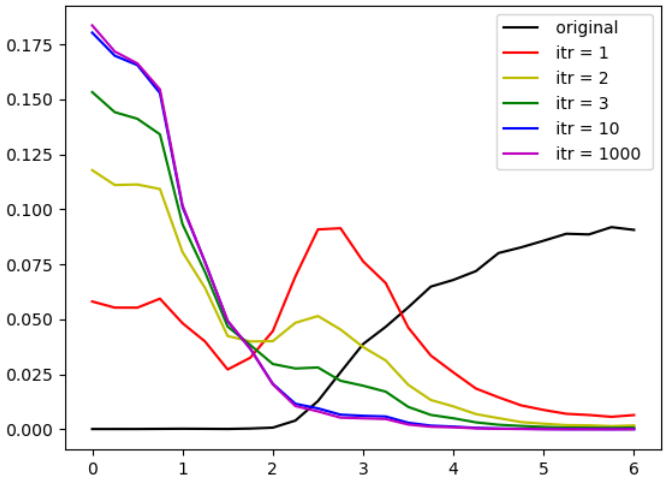
- Configurations at each temp have a characteristic scale of length, and the RBM seems to grasp it as a feature.
- We can check it as follows: If we train the RBM using only configs at specific temperature, the flow of temp approaches *that* temperature.





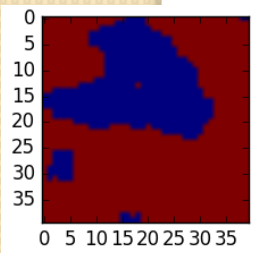


RBM learning T=0, …, 6

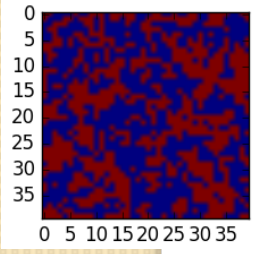


RBM learning only T=0

- Configs at T=0, …, 6 have different scales from each other. If the RBM learns these configs simultaneously, it may try to grasp all these scales as a feature of these configs.



- As a result, the RBM may learn the scale invariant configs as a feature of all the configs, since such configs have various different characteristic scales.



- As you know, the configs at the critical temperature T=2.27 do have scale invariance. This must be why the RBM flow approach the critical temp.
- This is our conjecture. We want to find more supporting evidences by analyzing the weight matrix of the RBM.

Analysis on WW^T

- When we input v_i for visible layer, the probability that an output of hidden layer is h_a (if ignoring bias):

$$p(\{h_a\}) = \frac{\exp \sum_i v_i w_{ia} h_a}{2 \cosh \sum_i v_i w_{ia}} \quad \leftarrow \quad B_a := \sum_i v_i w_{ia}$$

can be regarded as external field.

- When we input h_a for hidden layer, the probability that an output of visible layer is $\tilde{v}_i$:

$$\tilde{p}(\{\tilde{v}_i\}) = \frac{\exp \sum_a \tilde{v}_i w_{ia} h_a}{2 \cosh \sum_a w_{ia} h_a} \quad \leftarrow$$

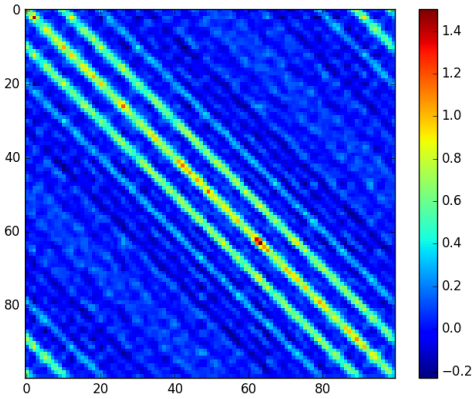
Substitute h_a for B_a, since they tend to correlate.

$$\rightarrow \frac{\exp \sum_a \tilde{v}_i w_{ia} \sum_j w_{ja} v_j}{2 \cosh \sum_a w_{ia} \sum_j w_{ja} v_j} = \frac{\exp \sum_j \tilde{v}_i (ww^T)_{ij} v_j}{2 \cosh \sum_j (ww^T)_{ij} v_j}$$

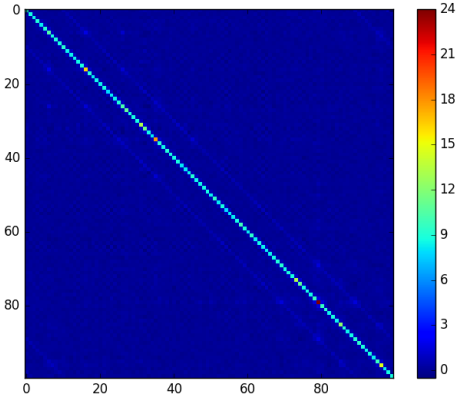
- ww^T must have important information about learning (which is independent from changes of basis in hidden layer).

- Elements of ww^T matrix

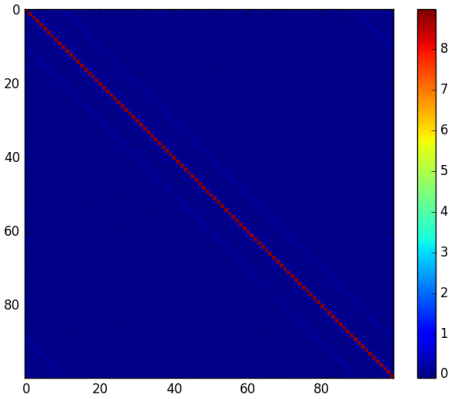
100 (input v) x 100 (output $\tilde{v}$) matrix



Hidden: $n_h = 16$
It cannot be diagonal due to small dofs of the weight matrix.



Hidden: $n_h = 100 = n_v$
It is close to be diagonal.

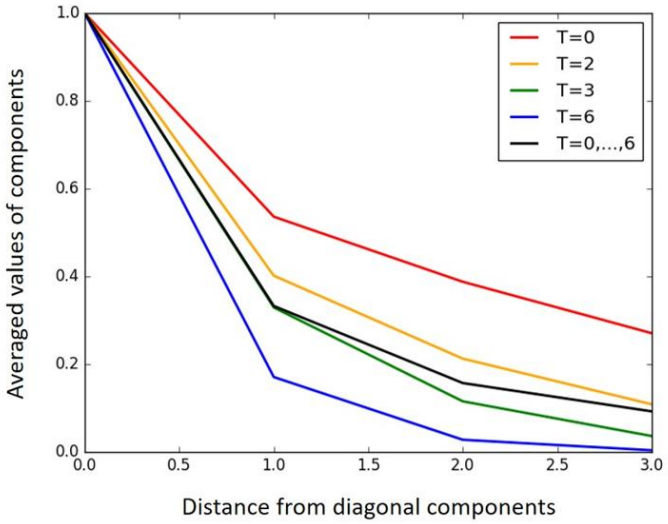


Hidden: $n_h = 400$
It is almost diagonal.

to make $\tilde{v}_i$ approach v_i

- In the case of $n_h = 16$, damping from diag elements shows a "characteristic scale" of configurations at each temp the RBM has learned.

The RBM grasps a feature!



- Singular value decomposition of ww^T matrix

- Expand an input config v in a linear combination of eigenvectors.

$$v = \sum_a c_a u_a$$

$$ww^T u_a = \lambda_a u_a$$

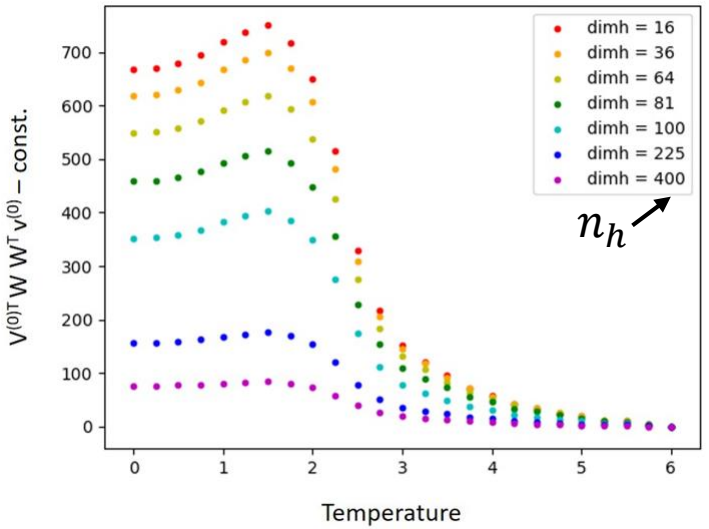
- Act the vector v on the ww^T matrix from left and right.

$$v^T ww^T v = \sum_a c_a^2 \lambda_a$$

It becomes large if including large principle components of ww^T.

- The averaged values of 1000 input configs at each temp shows a great change around the critical temp (T=2.27). It behaves like magnetization.

- This shows that the RBM did learn the critical temp, though only spin configs are given.

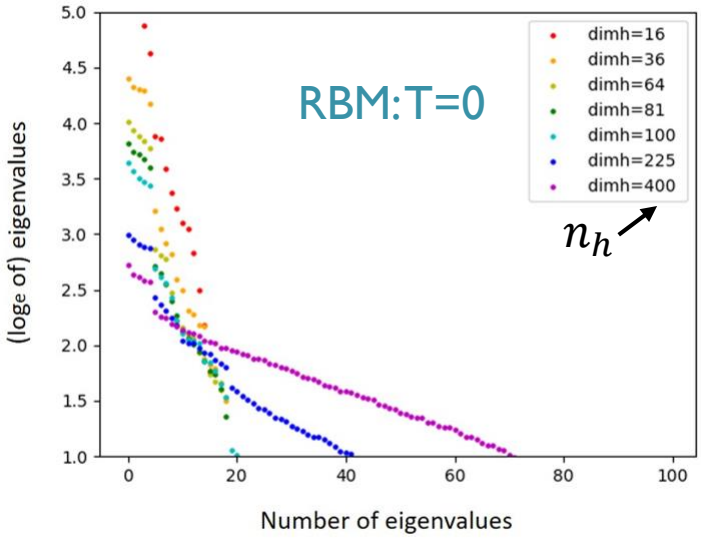


- Spectrum of eigenvalues

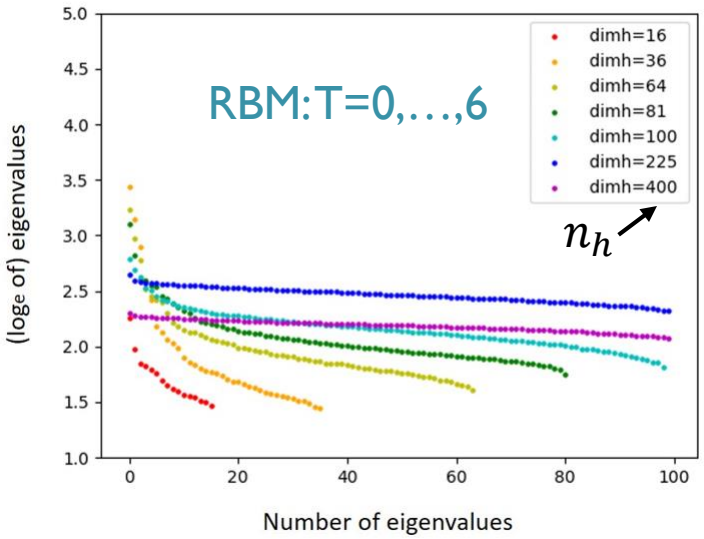
$$ww^T u_a = \lambda_a u_a$$

- If the RBM learns configs at only specific temperature, only a few (about five) eigenvalues are especially large.
- If the RBM learns all the configs at T=0, 0.25, …, 6, all the eigenvalues have similar values. (It roughly shows the scale invariance.) This may be because many hidden neurons are necessary to learn various scales at various temperatures.

$\log|\lambda_a|$

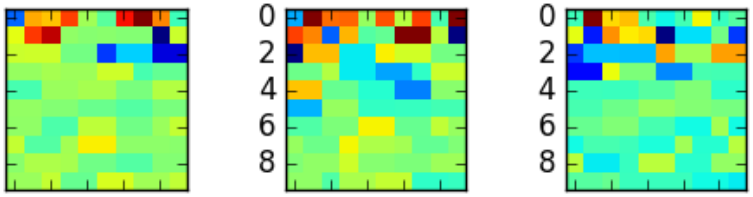


$\log|\lambda_a|$

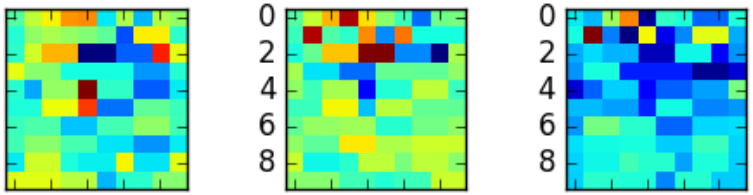


- What can we find from eigenvectors…? $ww^T u_a = \lambda_a u_a$
  - Principle components of ww^T in RBM learning only T=0,…,2

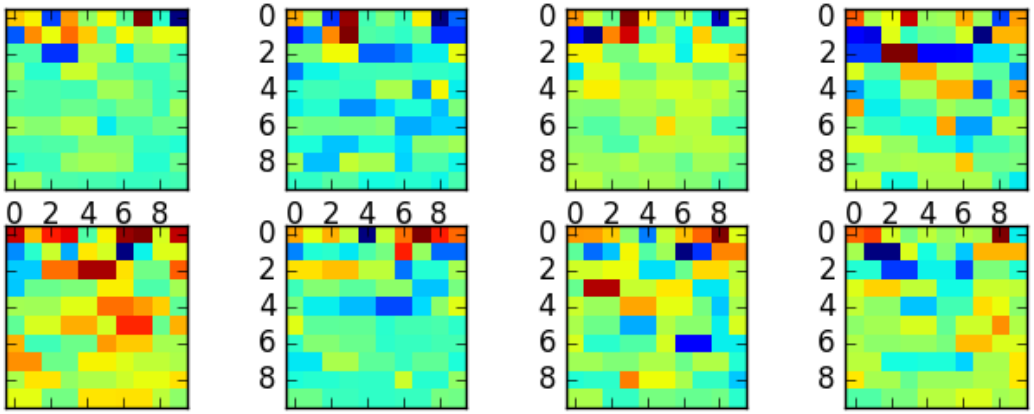
Eigenvectors with large eigenvalues



  - Those of RBM learning only T=4, …, 6



  - Those of RBM learning all T=0,…,6



At present, we cannot grasp any meaningful information from eigenvectors…

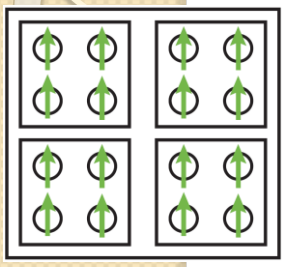
Summary & future directions

- We perform a machine learning of the RBM using images of spin configurations in 2d Ising model.
- We find the flow of reconstructed images by RBM is similar to the RG flow, but they are in the opposite directions.
- We conjecture that if the RBM learns images with various length scale, it grasps (or extracts) a scale invariant image as a feature of all these images.
- In future works, we will discuss whether our conjecture is correct in more general systems.
- We are going to study Blume-Capel model, where the 1st order phase transition occurs as well as the 2nd order.
- Is the RBM flow related to the way of human recognitions…?

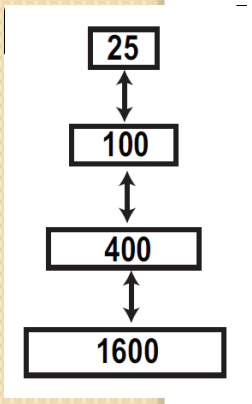
Backup

Naïve expectation: relation to RG

[Mehta-Schwab, '14]

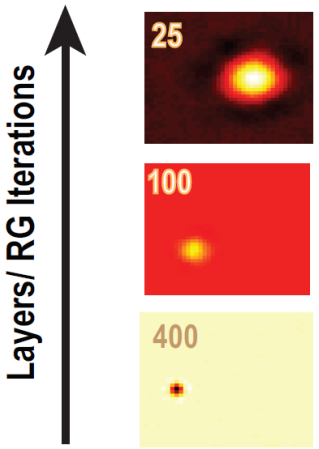


- The RBM may correspond to block spin transformation.
- There is some previous research using 2d Ising model:
- First we input images of 40x40 configs to visible layer, and set the size of hidden layer 20x20, then the weight matrix (1600x400) learn them.



- After the learning, we get output images of 20x20 configs from hidden layer.
- Next we input the 20x20 configs to visible layer, and set size of hidden layer 10x10, then weights (400x100) learn them.
- Next we input the 10x10 configs to visible layer, set size of hidden layer 5x5, …

- Looking at weight matrix after learning, …
  - We find that if visible and hidden layers have smaller size, the RBM has a wider "field of view" to recognize images.
  - Here the "field of view" means the region where elements of weight matrix has relatively large absolute values.



  - A kind of "coarse graining" seems to occur here.
    This is a similar phenomenon to the renormalization flow.

- On the other hand, some papers claim that the RBM is not related to RG. [Lin-Tegmark-Rolnick, '16]
  However, they seem not to exhibit concrete evidence.

Goals for machine learning

- このような技術を応用して、ビジネスチャンスを作ろうという動きが活発になっている。
  - 画像認識を用いた、車の自動運転技術の精度向上など。
  - 同様のNNを用いた、音声認識の技術。自然言語処理に応用した、文章の要約や翻訳の技術。
- アカデミズムの人間としては、産業界がやらない部分で成果を出したい。
  - なぜ機械学習がうまくいくのか、**仕組みを解明**したい。
  - 乳児が世界を認識していくプロセスに似ているという期待もある。意識の仕組みの解明に繋がるか？