Boltzmann Machine

CS385 Machine Learning – Artificial Neural Network

Remembering process with SA (Rev.)

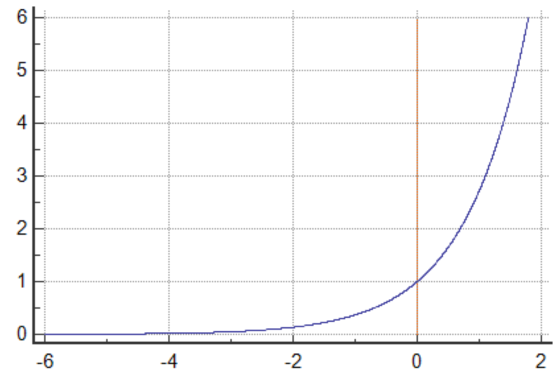
- HNN, 3 nodes, the status of each neuron is either 0 or 1
- consider neuron 1, its status S1 is 1, to determine if it changed to S1'(0)

$$\Delta E = Enew - Eold = -(S1'*S2*w12 + S1'*S3*w13 + S2*S3*w23) + (S1*S2*w12 + S1*S3*w13 + S2*S3*w23)$$

$$= (S1-S1')(S2*w12+S3*w13\ )$$

$$= (S2*w12+S3*w13)$$

$\exp(-\Delta E/T)$>threshold accept



Boltzmann distribution

- HNN, 3 nodes, the status of each neuron is either 0 or 1
- consider neuron 1, its status S1 is 1, to determine if it changed to S1’(0)

- Boltzmann distribution: $F(config) \propto \exp(-\frac{E}{kT})$

  F(new) = exp(-Enew/T)

  F(old) = exp(-Eold/T)

- Boltzmann factor

  F(new)/F(old)= exp(-ΔE/T)

A probabilistic view

- HNN, 3 nodes, the status of each neuron is either 0 or 1
- consider neuron 1, its status S1 is 1, to determine if it changed to S1'(0)

p(S1=0)=F(new) = exp(-Enew/T) → Enew = -T·ln(p(S1=0))

p(S1=1)=F(old) = exp(-Eold/T) → Eold = -T·ln(p(S1=1))

ΔE = Enew – Eold = -T·ln(p(S1=0)) + T·ln(p(S1=1))

$$\frac{\Delta E}{T} = \ln(p(S1 = 1) - \ln\big(1 - p(S1 = 1)\big) = \ln\left(\frac{p(s1 = 1)}{1 - p(s1 = 1)}\right)$$

A probabilistic view – stochastic unit

- HNN, 3 nodes, the status of each neuron is either 0 or 1
- consider neuron 1, its status S1 is 1, to determine if it changed to S1'(0)

$$\frac{\Delta E}{T} = \ln\left(\frac{p(s1=1)}{1-p(s1=1)}\right)$$

$$-\frac{\Delta E}{T} = \ln\left(\frac{1-p(s1=1)}{p(s1=1)}\right) = \ln\left(\frac{1}{p(s1=1)} - 1\right)$$

$$\exp\left(-\frac{\Delta E}{T}\right) = \frac{1}{p(s1=1)} - 1 \quad \rightarrow \quad p(s1=1) = \frac{1}{1+\exp\left(-\frac{\Delta E}{T}\right)}$$

(neuron & Energy) can be used to compute prob.

sigmoid $\left(-\frac{\Delta E}{T}\right)$

Thermal equilibrium

- Set temperature =1
- Thermal equilibrium is a difficult concept!
  - Reaching thermal equilibrium does not mean that the system has settled down into the lowest energy configuration.
  - The thing that settles down is the probability distribution over configurations.
  - This settles to the stationary distribution.
- A Boltzmann machine is a model which describes data distribution

An analogy

- Imagine a casino in Sentosa that is full of card dealers (we need many more than 52! of them).
- We start with all the card packs in standard order and then the dealers all start shuffling their packs.
  - After a few time steps, the king of spades still has a good chance of being next to the queen of spades. The packs have not yet forgotten where they started.
  - After prolonged shuffling, the packs will have forgotten where they started. There will be an equal number of packs in each of the 52! possible orders.
  - Once equilibrium has been reached, the number of packs that leave a configuration at each time step will be equal to the number that enter the configuration.
- The only thing wrong with this analogy is that all the configurations have equal energy, so they all end up with the same probability.

5 cards

$5 \times 4 \times 3 \times 2 \times 1 = 120$ orders

120 card packs

reach equilibrium

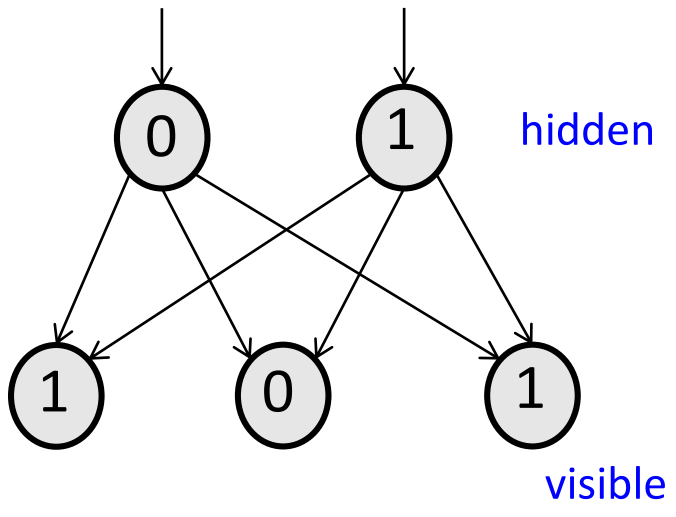
s	P(s)
(1 2 3 4 5)	$\frac{1}{120}$
(1 2 3 5 4)	$\frac{1}{120}$
⋮	⋮
(5 4 3 2 1)	$\frac{1}{120}$

$\longrightarrow$ shuffle one more time simultaneously $\longrightarrow$

s	P(s)
(1 3 4 5 2)	$\frac{1}{120}$
(1 2 3 4 5)	$\frac{1}{120}$
⋮	⋮
(1 2 3 5 4)	$\frac{1}{120}$

stationary distribution

Boltzmann machine



- A BM has two layers: hidden and visible
- RBM: constraint connectivity,
  - only connect hidden layer and visible layer
- The energy of a joint configuration

binary state of unit i in **v**

bias of unit k

RBM

RBM

$$-E(\mathbf{v},\mathbf{h}) = \sum_{i \in vis} v_i b_i + \sum_{k \in hid} h_k b_k + \sum_{i<j} v_i v_j w_{ij} + \sum_{i,k} v_i h_k w_{ik} + \sum_{k<l} h_k h_l w_{kl}$$

Energy with configuration **v** on the visible units and **h** on the hidden units

indexes every non-identical pair of i and j once

weight between visible unit i and hidden unit k

Energy → probability

- v: observable features
- h: cluster/class

v: (headach = 1, sneeze = 0)

h: (concussion=1, cold=0)

$E(10, 10)$

$$p(\mathbf{v},\mathbf{h}) = \frac{e^{-\underline{E(\mathbf{v},\mathbf{h})}}}{\sum_{\mathbf{u},\mathbf{g}} e^{-E(\mathbf{u},\mathbf{g})}}$$

partition function

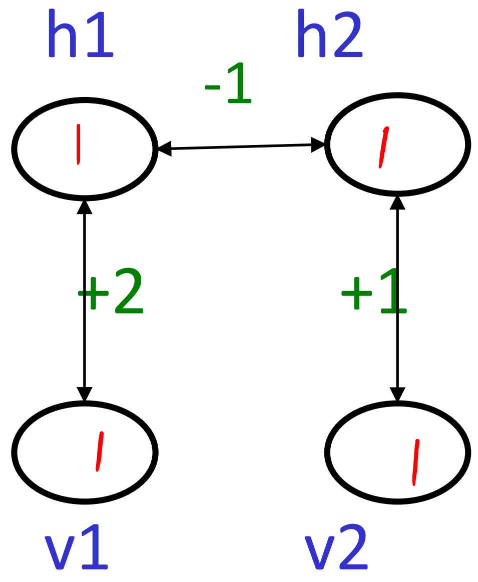
$$\begin{array}{cccc} u & & g & \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ & \vdots & & \\ 1 & 1 & 1 & 1 \end{array} \quad 2^4$$

$$p(\mathbf{v}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{u},\mathbf{g}} e^{-E(\mathbf{u},\mathbf{g})}}$$

An example of how weights define a distribution

$\mathbf{v}$	$\mathbf{h}$	$-E$	e^{-E}	$p(\mathbf{v}, \mathbf{h})$	$p(\mathbf{v})$
1 1	1 1	2	7.39	.186 $= \frac{7.39}{39.7}$	0.466 $= \frac{2\times 7.39 + 2.72 + 1}{39.7}$
1 1	1 0	2	7.39	.186	
1 1	0 1	1	2.72	.069	
1 1	0 0	0	1	.025	
1 0	1 1	1	2.72	.069	0.305
1 0	1 0	2	7.39	.186	
1 0	0 1	0	1	.025	
1 0	0 0	0	1	.025	
0 1	1 1	0	1	.025	0.144
0 1	1 0	0	1	.025	
0 1	0 1	1	2.72	.069	
0 1	0 0	0	1	.025	
0 0	1 1	-1	0.37	.009	0.084
0 0	1 0	0	1	.025	
0 0	0 1	0	1	.025	
0 0	0 0	0	1	.025	

$(\sum e^{-E}) = 39.70$



$-E = 1\times 1\times -1 + 1\times 1\times 2 + 1\times 1\times 1$

Classify a new vector

- Once the weights are learnt

- Let the visible units clamped to the given new vector
- Chang the status for each hidden unit

- Better explanations/classes have low energy/higher probability

MLE with BM

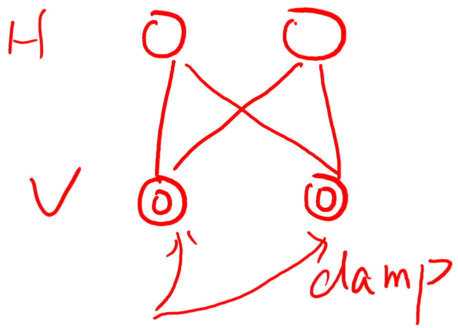
- To maximize the product of the probabilities that the Boltzmann machine assigns to the binary vectors in the training set.

$$\frac{\partial \log p(\mathbf{v})}{\partial w_{ij}} = \langle s_i s_j \rangle_{\mathbf{v}} - \langle s_i s_j \rangle_{model}$$

Derivative of log probability of one training vector, v under the model.

Expected value of product of states at thermal equilibrium when v is clamped on the visible units

Expected value of product of states at thermal equilibrium with no clamping



< o o > in training data

$$\Delta w_{ij} \propto \langle s_i s_j \rangle_{data} - \langle s_i s_j \rangle_{model}$$

An inefficient way to collect the statistics required for learning

Hinton and Sejnowski (1983)

- Positive phase: Clamp a data vector on the visible units and set the hidden units to random binary states.
  - Update the hidden units one at a time until the network reaches thermal equilibrium at a temperature of 1.
  - Sample $< s_i s_j >$ for every connected pair of units.
  - Repeat for all data vectors in the training set and average.

V: s_1	s_2	H: s_3	s_4	$\langle s_1 s_3 \rangle$
0	0	1	1	0
0	1	1	0	0
1	1	1	1	1
0	0	1	1	0

return $\frac{1}{4}$

- Negative phase: Set all the units to random binary states.
  - Update all the units one at a time until the network reaches thermal equilibrium at a temperature of 1.
  - Sample $< s_i s_j >$ for every connected pair of units.
  - Repeat many times (how many?) and average to get good estimates.

V: s_1	s_2	H: s_3	s_4	$\langle s_1 s_3 \rangle$
0	0	1	0	0
0	0	0	1	0
0	0	1	1	0
1	1	0	0	0
1	0	1	0	1
1	1	1	0	1

return $\frac{1}{3}$

$\Delta W_{13} = \frac{1}{4} - \frac{1}{3}$

Getting a sample from the model

- Run the machine(Markov chain) until it reaches its stationary distribution (thermal equilibrium at a temperature of 1).
  - Similar to the remembering process in HNN with Simulated Annealing
- The probability of a global configuration is then related to its energy by the Boltzmann distribution

$$p(\mathbf{v},\mathbf{h}) \propto e^{-E(\mathbf{v},\mathbf{h})}$$