Lecture 12 Quiz

Continue Course

Back to Week 12

7/7 points earned (100%)

1 / 1 points

1.
The Boltzmann Machine learning algorithm involves computing two expectations -

- $\langle s_i s_j \rangle_{data}$: Expected value of $s_i s_j$ at equilibrium when the visible units are fixed to be the data.
- $\langle s_i s_j \rangle_{model}$: Expected value of $s_i s_j$ at equilibrium when the visible units are not fixed.

When applied to a general Boltzmann Machine (not a Restricted one), this is an approximate learning algorithm because

- [x] There is no efficient way to compute the first expectation exactly.

Correct
Computing $\langle s_i s_j \rangle_{data}$ is hard in general. It usually involves sampling from the model conditioned on the data.

- [ ] The first expectation can be computed exactly, but the second one cannot be.

Un-selected is correct

- [x] There is no efficient way to compute the second expectation exactly.

Correct

Computing $\langle s_i s_j \rangle_{model}$ is hard in general. It usually involves sampling from the model.

- [x] The first expectation cannot be computed exactly, but the second one can be.

Un-selected is correct

---

1 / 1 points

2.

Throughout the lecture, when talking about Boltzmann Machines, why do we talk in terms of computing the **expected** value of $s_i s_j$ and not the value of $s_i s_j$?

- It is not possible to compute the exact value no matter how much computation time is provided. So all we can do is compute an approximation.
- The expectation only refers to an average over all training cases.
- It is possible to compute the exact value but it is computationally inefficient.
- It does not make sense to talk in terms of a unique value of $s_i s_j$ because s_i and s_j are random variables and the Boltzmann Machine defines a probability distribution over them.

Correct

---

1 / 1 points

3.

When learning an RBM, we decrease the energy of data particles **and** increase the energy of fantasy particles. Brian insists that the latter is not needed. He claims that it is should be sufficient to just decrease the energy of data particles and the energy of all other regions of state space would have increased relatively. This would also save us the trouble of sampling from the model distribution. What is wrong with this intuition ?

- Since total energy is constant, some particles must loose energy for others to gain energy.
- The sum of all updates must be zero so we need to increase the energy of negative particles to balance things out.
- There is nothing wrong with the intuition. This method is an alternative way of learning a Boltzmann Machine.
- The model could decrease the energy of data particles in ways such that the energy of negative particles also gets decreased. If this happens there will be no net learning and energy of all particles will keep going down without bounds.

Correct
The network might update its weights to lower the energy of large regions of space surrounding the data particles and these regions and the decrease could be as large as possible. Another way to see this is to look at the weight update equation and notice that the gradient would never go to zero unless there are negative particles.

✔ 1 / 1 points

4.
Restricted Boltzmann Machines are easier to learn than Boltzmann Machines with arbitrary connectivity. Which of the following is a contributing factor ?

- The energy of any configuration of an RBM is a linear function of its states. This is not true for a general BM.
- RBMs are more powerful models, i.e., they can model more probability distributions than general BMs.
- In RBMs, there are no connections among hidden units or among visible units.

Correct

This makes it possible to update all hidden units in parallel given the visible units (and vice-versa). Moreover, only one such update gives the exact value of the expectation that is being computing.

- It is possible to run a persistent Markov chain in RBMs but not in general BMs.

---

1 / 1 points

5.

PCD a better algorithm than CD1 when it comes to training a good generative model of the data. This means that samples drawn from a freely running Boltzmann Machine which was trained with PCD (after enough time) are likely to look more realistic than those drawn from the same model trained with CD1. Why does this happen ?

- In PCD, many Markov chains are used throughout learning, whereas CD1 uses only one. Therefore, samples from PCD are an average of samples from several models. Since model averaging helps, PCD generates better samples.
- In PCD, the persistent Markov chain can remember the state of the positive particles across mini-batches and show them when sampling. However, CD1 resets the Markov chain in each update so it cannot retain information about the data for a long time.
- In PCD, only a single Markov chain is used throughout learning, whereas CD1 starts a new one in each update. Therefore, PCD is a more consistent algorithm.
- In PCD, the persistent Markov chain explores different regions of the state space. However, CD1 lets the Markov chain run for only one step. So CD1 cannot explore the space of possibilities much and can miss out on increasing the energy of some states which ought to be improbable. These states might be reached when running the machine for a long time leading to unrealistic samples.

Correct

---

1 / 1 points

6.
It's time for some math now!

In RBMs, the energy of any configuration is a linear function of the state.

$$E(\mathbf{v}, \mathbf{h}) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_j W_{ij}$$

and this eventually leads to

$$\Delta W_{ij} \propto \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}$$

If the energy was non-linear, such as

$$E(\mathbf{v}, \mathbf{h}) = -\sum_i a_i f(v_i) - \sum_j b_j g(h_j) - \sum_{i,j} f(v_i) g(h_j) W_{ij}$$

for some non-linear functions f and g, which of the following would be true.

- $\Delta W_{ij} \propto f(\langle v_i \rangle_{data}) g(\langle h_j \rangle_{data}) - f(\langle v_i \rangle_{model}) g(\langle h_j \rangle_{model})$
- $\Delta W_{ij} \propto \langle f(v_i) g(h_j) \rangle_{data} - \langle f(v_i) g(h_j) \rangle_{model}$

Correct

$$\begin{aligned} p(v) &= \exp(-E(\mathbf{v}, \mathbf{h}))/Z \\ \Rightarrow \log(p(\mathbf{v})) &= -E(\mathbf{v}, \mathbf{h}) - \log(Z) \\ \Rightarrow \frac{\partial \log(p(\mathbf{v}))}{\partial W_{ij}} &= f(v_i) g(h_j) - \sum_{\mathbf{v}', \mathbf{h}'} P(v_i', h_j') f(v_i') g(h_j') \end{aligned}$$

Averaging over all data points,

$$\begin{aligned} \frac{\partial \log(p(\mathbf{v}))}{\partial W_{ij}} &= \langle f(v_i) g(h_j) \rangle_{data} - \langle f(v_i) g(h_j) \rangle_{model} \\ \Delta W_{ij} &\propto \frac{\partial \log(p(\mathbf{v}))}{\partial W_{ij}} \\ \Rightarrow \Delta W_{ij} &\propto \langle f(v_i) g(h_j) \rangle_{data} - \langle f(v_i) g(h_j) \rangle_{model} \end{aligned}$$

- $\Delta W_{ij} \propto \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}$
- $\Delta W_{ij} \propto \langle f(v_i) \rangle_{data} \langle g(h_j) \rangle_{data} - \langle f(v_i) \rangle_{model} \langle g(h_j) \rangle_{model}$

1 / 1 points

7.

In RBMs, the energy of any configuration is a linear function of the state.

and this eventually leads to

If the energy was non-linear, such as

for some non-linear function , which of the following would be true.

- None of these is correct.
- $P(h_j = 1|\mathbf{v}) = \frac{1}{1+\exp(-\sum_i W_{ij}(f(v_i,1)+f(v_i,0))-b_j)}$
- $P(h_j = 1|\mathbf{v}) = \frac{1}{1+\exp(-\sum_i W_{ij}v_i-b_j)}$
- $P(h_j = 1|\mathbf{v}) = \frac{1}{1+\exp(-\sum_i W_{ij}(f(v_i,1)-f(v_i,0))-b_j)}$

Correct