Feedback — Lecture 14 Quiz

Help Center

You submitted this quiz on **Fri 1 Jul 2016 2:03 AM CEST**. You got a score of **6.00** out of **8.00**. However, you will not get credit for it, since it was submitted past the deadline.

Question 1

Why is a Deep Belief Network not a Boltzmann Machine ?

Your Answer		Score	Explanation
○ A DBN is not a probabilistic model of the data.			
◉ Some edges in a DBN are directed.	✔	1.00	In a Boltzmann Machine, all edges must be undirected. A DBN has directed edges from the top-level RBM to each subsequent layer below.
○ A DBN does not have hidden units.			
○ All edges in a DBN are directed.			
Total		1.00 / 1.00	

Question 2

Brian looked at the direction of arrows in a DBN and was surprised to find that the data is at the "output". "Where is the input ?!", he exclaimed, "How will I give input to this model and get all those cool features?" In this context, which of the following statements are true? Check all that apply.

Your Answer		Score	Explanation
☐ A DBN is a generative model of the data and cannot be used to generate features for any	✔	0.50	This is not true. A DBN can be used to generate features for any given data vector by doing inference.

given input. It can only be used to get features for data that was generated by the model.			
☐ A DBN is a generative model of the data, which means that, its arrows define a way of generating data from a probability distribution, so there is no "input".	✖	0.00	Unlike a feed-forward neural net, a DBN is not a mapping from inputs to outputs. It is a probabilistic generative model of the data.
☐ In order to get features h given some data v, he must perform inference to find out $P(h\|v)$. There is an easy **approximate** way of doing this, just traverse the arrows in the opposite direction.	✖	0.00	Traversing arrows in the opposite direction is an approximate inference procedure.
☐ In order to get features h given some data v, he must perform inference to find out $P(h\|v)$. There is an easy **exact** way of doing this, just traverse the arrows in the opposite direction.	✔	0.50	Traversing arrows in the opposite direction is an approximate inference procedure.
Total		1.00 / 2.00	

Question 3

In which of the following cases is pretraining likely to help the most (compared to training a neural net from random initialization) ?

Your Answer	Score	Explanation
○ A dataset of images is to be classified into 100 semantic classes. Fortunately, there are 100 million labelled training examples.		
○ A dataset of binary pixel images which are to be classified based on		

parity, i.e., if the sum of pixels is even the image has label 0, otherwise it has label 1.			
○ A speech dataset with 10 billion labelled training examples.			
◉ A dataset of images is to be classified into 100 semantic classes. There are only 1,000 labelled images but 100 million unlabelled ones are available from the internet.	✔	1.00	The small labelled set of images is unlikely to have enough information to learn good features. The unlabelled images can be used to do pretraining for learning features.
Total		1.00 / 1.00	

Question 4

Why does pretraining help more when the network is deep ?

Your Answer		Score	Explanation
☐ As nets get deeper, contrastive divergence objective used during pretraining gets closer to the classification objective.	✔	0.50	Contrastive divergence has nothing to do with the classification objective.
☑ During backpropagation in very deep nets, the lower level layers get **very small gradients**, making it hard to learn good low-level features. Since pretraining starts those low-level	✔	0.50	Lower level layers can get very small gradients, especially if saturating hidden units are used (such as logistic or tanh units). Pretraining can initialize the weights in a proper region of weight space so that the features don't have to start learning from scratch.

features off at a good point, there is a big win.			
☐ Deeper nets have more parameters than shallow ones and they overfit easily. Therefore, initializing them sensibly is important.	✖	0.00	More parameters means that the model can find ingenious ways of overfitting by learning features that don't generalize well. Pretraining can initialize the weights in a proper region of weight space so that the features learned are not too bad.
☐ Backpropagation algorithm cannot give accurate gradients for very deep networks. So it is important to have good initializtions, especially, for the lower layers.	✔	0.50	Backpropgation gives exact gradients.
Total		1.50 / 2.00	

Question 5

The energy function for binary RBMs goes by

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

When modeling real-valued data (i.e., when $\mathbf{v}$ is a real-valued vector not a binary one) we change it to

$$E(\mathbf{v}, \mathbf{h}) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_j h_j a_j - \sum_{i,j} \frac{v_i}{\sigma_i} W_{ij} h_j$$

Why can't we still use the same old one ?

Your Answer		Score	Explanation
☐ If we use the old one, the real-valued vectors would end up being constrained to be binary.	✔	0.50	The energy function does not impose any such constraints.

☐ Probability distributions over real-valued data can only be modeled by having a conditional Gaussian distribution over them. So we have to use a quadratic term.	✔	0.50	This is not true. Distributions over real-valued data can be modelled by a multitude of distributions or combinations thereof. The choice of Gaussian distribution is arbitrary.
☑ If the model assigns an energy e_1 to state $\mathbf{v_1, h}$, and e_2 to state $\mathbf{v_2, h}$, then it would assign energy $(e_1 + e_2)/2$ to state $\mathbf{(v_1 + v_2)/2, h}$. This does not make sense for the kind of distributions we usually want to model.	✔	0.50	Suppose v_1 and v_2 represent two images. We would like e_1 and e_2 to be small. This makes the energy of the average image low, but the average of two images would not look like a natural image and should not have low energy.
☐ If we continue to use the same one, then in general, there will be infinitely many $\mathbf{v}$'s and $\mathbf{h}$'s such that, $E(\mathbf{v, h})$ will be infinitely small (close to $-\infty$). The probability distribution resulting from such an energy function is not useful for modeling real data.	✖	0.00	If some $b_i < 0$, then if $v_i \to -\infty$, then $E \to -\infty$. Similarly for $b_i > 0$ if $v_i \to \infty$, then $E \to -\infty$. So the Boltzmann distribution based on this energy function would behave in weird ways.
Total		1.50 / 2.00	