Large Scale Machine Learning and Deep Learning
Review Questions 8

1. Assume we have a stacked autoencoder with three hidden layers $\mathtt{h}_1$, $\mathtt{h}_2$, and $\mathtt{h}_3$, in which each layer applies the following functions respectively, $\mathbf{h_1} = \mathtt{f}_1(\mathbf{x})$, $\mathbf{h_2} = \mathtt{f}_2(\mathbf{h}_1)$, and $\mathbf{h}_3 = \mathtt{f}_3(\mathbf{h}_2)$, and the output of the network will be $\mathbf{y} = \mathtt{f}_4(\mathbf{h}_3)$. Do you think if it is a good autoencoder if it generates $\mathtt{f}_4(\mathtt{f}_3(\mathtt{f}_2(\mathtt{f}_1(\mathbf{x})))) = \mathbf{x}$ for all input instances $\mathbf{x}$. How can we improve it?

   Answer: the Autoencoder slides, pages 12–14, 23–29

---

2. How does Gibbs sampling work? When do we need to use Gibbs sampling?

   Answer: Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm for obtaining a sequence of observations that are approximated from a specified multivariate probability distribution, when direct sampling is difficult. This sequence can be used to approximate the joint distribution. The point of Gibbs sampling is that given a multivariate distribution, it is simpler to sample from a conditional distribution than to marginalize by integrating over a joint distribution. Suppose $\mathtt{p}(\mathtt{x}, \mathtt{y})$ is a probability mass function (PMF) that is difficult to sample from directly, but assume we can easily sample from the conditional distributions $\mathtt{p}(\mathtt{x} \mid \mathtt{y})$ and $\mathtt{p}(\mathtt{y} \mid \mathtt{x})$. The Gibbs sampler first sets $(\mathtt{x}_0, \mathtt{y}_0)$ to some initial value, and then repeats the following iterations $\mathtt{k}$ times:

   1. Sample $\mathtt{x}_1 \sim \mathtt{p}(\mathtt{x} \mid \mathtt{y}_0) \Rightarrow$ current state: $(\mathtt{x}_1, \mathtt{y}_0)$
      Sample $\mathtt{y}_1 \sim \mathtt{p}(\mathtt{y} \mid \mathtt{x}_1) \Rightarrow$ current state: $(\mathtt{x}_1, \mathtt{y}_1)$
   2. Sample $\mathtt{x}_2 \sim \mathtt{p}(\mathtt{x} \mid \mathtt{y}_1) \Rightarrow$ current state: $(\mathtt{x}_2, \mathtt{y}_1)$
      Sample $\mathtt{y}_2 \sim \mathtt{p}(\mathtt{y} \mid \mathtt{x}_2) \Rightarrow$ cCurrent state: $(\mathtt{x}_2, \mathtt{y}_2)$

---

3. What is a generative model? Can you name a type of generative autoencoder?

   Answer: a generative model is a model capable of randomly generating outputs that resemble the training instances. For example, once trained successfully on the MNIST dataset, a generative model can be used to randomly generate realistic images of digits. The output distribution is typically similar to the training data. For example, since MNIST contains many images of each digit, the generative model would output roughly the same number of images of each digit. Some generative models can be parametrized, for example, to generate only some kinds of outputs. An example of a generative autoencoder is the variational autoencoder.

4. How do you tie weights in a stacked autoencoder? What is the point of doing so?

   Answer: to tie the weights of an encoder layer and its corresponding decoder layer, you make the decoder weights equal to the transpose of the encoder weights. This reduces the number of parameters in the model by half, often making training converge faster with less training data, and reducing the risk of overfitting the training set.