Confronting the Partition Function

Lecture slides for Chapter 18 of *Deep Learning*

www.deeplearningbook.org

Ian Goodfellow

Last updated 2017-12-29

Unnormalized models

$$p(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})}\tilde{p}(\mathbf{x};\boldsymbol{\theta}). \tag{18.1}$$

where Z is

$$\int \tilde{p}(\boldsymbol{x})d\boldsymbol{x} \tag{18.2}$$

or

$$\sum_{\boldsymbol{x}} \tilde{p}(\boldsymbol{x}). \tag{18.3}$$

Gradient of log-likelihood

$$\nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}; \boldsymbol{\theta}) = \underbrace{\nabla_{\boldsymbol{\theta}} \log \tilde{p}(\mathbf{x}; \boldsymbol{\theta})}_{\text{Positive phase: push up on data points}} - \underbrace{\nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta})}_{\text{Negative phase: push down model samples}}. \tag{18.4}$$

Positive phase:
push up on data points

Negative phase:
push down model
samples

Negative phase sampling

$$\nabla_{\boldsymbol{\theta}} \log Z \tag{18.5}$$

$$= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \nabla_{\boldsymbol{\theta}} \log \tilde{p}(\mathbf{x}). \tag{18.13}$$

Basic learning algorithm for undirected models

- For each minibatch:
  - Generate model samples
  - Compute positive phase using data samples
  - Compute negative phase using model samples
  - Combine positive and negative phases, do a gradient step to update parameters

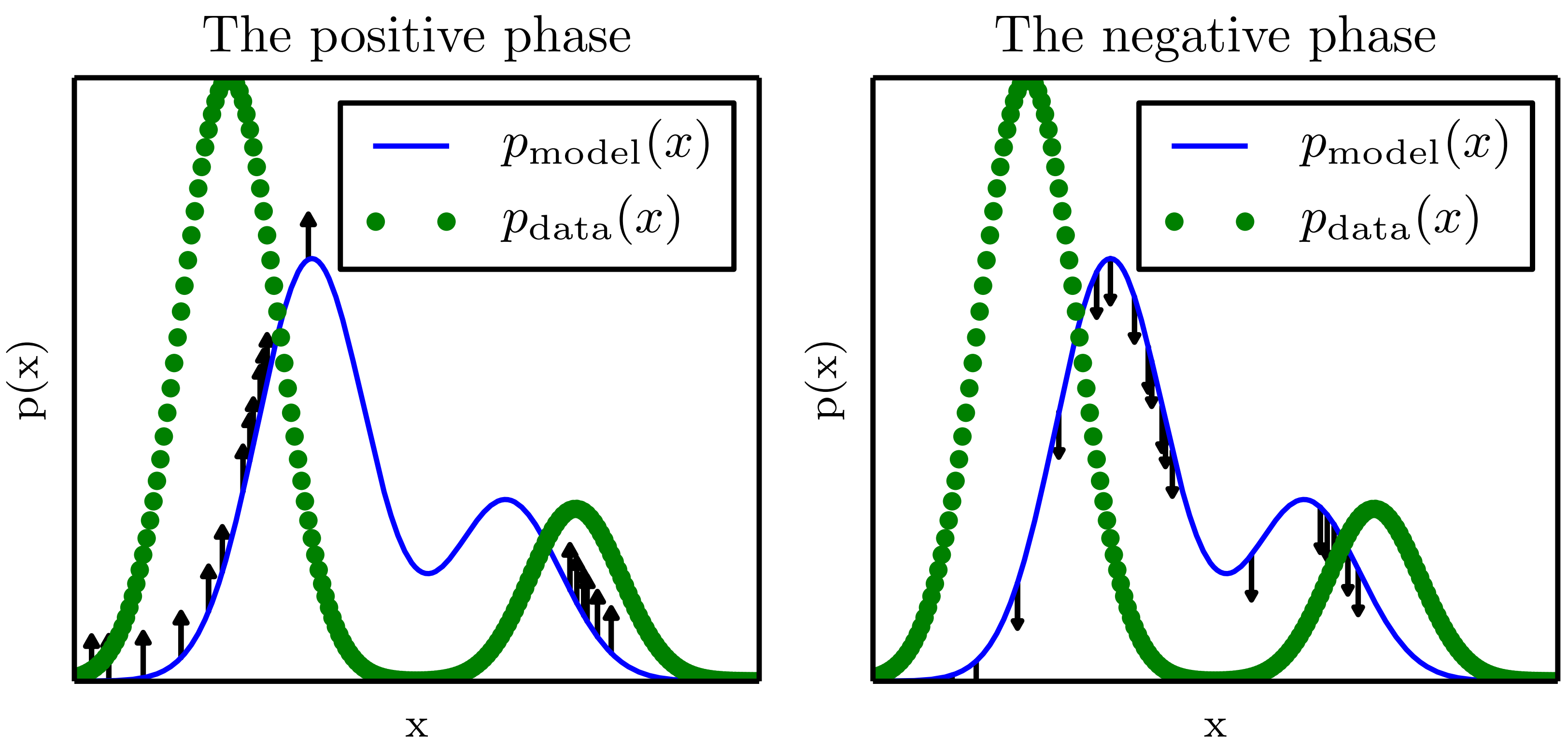

Figure 18.1: The view of algorithm 18.1 as having a "positive phase" and a "negative phase." *(Left)*In the positive phase, we sample points from the data distribution and push up on their unnormalized probability. This means points that are likely in the data get pushed up on more. *(Right)*In the negative phase, we sample points from the model distribution and push down on their unnormalized probability. This counteracts the positive phase's tendency to just add a large constant to the unnormalized probability everywhere. When the data distribution and the model distribution are equal, the positive phase has the same chance to push up at a point as the negative phase has to push down. When this occurs, there is no longer any gradient (in expectation), and training must terminate.

Challenge: model samples are slow

- Undirected models usually need Markov chains
- Naive approach: run the Markov chain for a long time starting from random initialization each minibatch
- Speed tricks:
    - Contrastive divergence: start the Markov chain from data
    - Persistent contrastive divergence: for each minibatch, continue the Markov chain from where it was for the previous minibatch

Sidestep the problem

- Use other criteria besides likelihood so that there is no need to compute Z or its gradient
    - Pseudolikelihood
    - Score matching
    - Ratio matching
    - Noise contrastive estimation

Estimating the Partition Function

- To evaluate a trained model, we want to know the likelihood
- This requires estimating Z, even if we trained using a method that doesn't differentiate Z
- Can estimate Z using *annealed importance sampling*