

training procedure presented in algorithm 18.1. The high cost of burning in the Markov chains in the inner loop makes this procedure computationally infeasible, but this procedure is the starting point that other more practical algorithms aim to approximate.

---

**Algorithm 18.1** A naive MCMC algorithm for maximizing the log-likelihood with an intractable partition function using gradient ascent.

---

Set  $\epsilon$ , the step size, to a small positive number.

Set  $k$ , the number of Gibbs steps, high enough to allow burn in. Perhaps 100 to train an RBM on a small image patch.

**while** not converged **do**

    Sample a minibatch of  $m$  examples  $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$  from the training set.

$\mathbf{g} \leftarrow \frac{1}{m} \sum_{i=1}^m \nabla_{\boldsymbol{\theta}} \log \tilde{p}(\mathbf{x}^{(i)}; \boldsymbol{\theta})$ .

    Initialize a set of  $m$  samples  $\{\tilde{\mathbf{x}}^{(1)}, \dots, \tilde{\mathbf{x}}^{(m)}\}$  to random values (e.g., from a uniform or normal distribution, or possibly a distribution with marginals matched to the model’s marginals).

**for**  $i = 1$  to  $k$  **do**

**for**  $j = 1$  to  $m$  **do**

$\tilde{\mathbf{x}}^{(j)} \leftarrow \text{gibbs\_update}(\tilde{\mathbf{x}}^{(j)})$ .

**end for**

**end for**

$\mathbf{g} \leftarrow \mathbf{g} - \frac{1}{m} \sum_{i=1}^m \nabla_{\boldsymbol{\theta}} \log \tilde{p}(\tilde{\mathbf{x}}^{(i)}; \boldsymbol{\theta})$ .

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \epsilon \mathbf{g}$ .

**end while**

---

We can view the MCMC approach to maximum likelihood as trying to achieve balance between two forces, one pushing up on the model distribution where the data occurs, and another pushing down on the model distribution where the model samples occur. Figure 18.1 illustrates this process. The two forces correspond to maximizing  $\log \tilde{p}$  and minimizing  $\log Z$ . Several approximations to the negative phase are possible. Each of these approximations can be understood as making the negative phase computationally cheaper but also making it push down in the wrong locations.

Because the negative phase involves drawing samples from the model’s distribution, we can think of it as finding points that the model believes in strongly. Because the negative phase acts to reduce the probability of those points, they are generally considered to represent the model’s incorrect beliefs about the world. They are frequently referred to in the literature as “hallucinations” or “fantasy particles.” In fact, the negative phase has been proposed as a possible explanation