Notes on Contrastive Divergence

Oliver Woodford

These notes describe Contrastive Divergence (CD), an approximate Maximum-Likelihood (ML) learning algorithm proposed by Geoffrey Hinton.

What is CD, and why do we need it?

Imagine that we would like to model the probability of a data point, x using a function of the form $f(x;\Theta)$, where Θ is a vector of model parameters. The probability of x, $p(x;\Theta)$ must integrate to 1 over all x, therefore:

$$p(x;\Theta) = \frac{1}{Z(\Theta)} f(x;\Theta) \tag{1}$$

where $Z(\Theta)$, known as the partition function, is defined as

$$Z(\Theta) = \int f(x;\Theta)\ dx \tag{2}$$

We learn our model parameters, Θ, by maximizing the probability of a training set of data, $\mathbf{X} = x_{1,..,K}$, given as

$$p(\mathbf{X};\Theta) = \prod_{k=1}^{K} \frac{1}{Z(\Theta)} f(x_k;\Theta) \tag{3}$$

or, equivalently, by minimizing the negative log of $p(\mathbf{X};\Theta)$, denoted $E(\mathbf{X};\Theta)$, which we shall call the energy:

$$E(\mathbf{X};\Theta) = \log Z(\Theta) - \frac{1}{K}\sum_{k=1}^{K} \log f(x_k;\Theta) \tag{4}$$

First, let us choose our probability model function, $f(x;\Theta)$, to be the pdf of a normal distribution, $\mathcal{N}(x;\mu,\sigma)$, so that $\Theta = \{\mu,\sigma\}$. The integral of the pdf is 1 (a standard result, though the proof is not trivial), so that $\log Z(\Theta) = 0$. Differentiating Equation 4 with respect to μ shows that the optimal μ is the mean of the training data, $\mathbf{X}$, and a similar calculation with respect to σ shows that the optimal σ is the square root of the variance of the training data.

Sometimes, as in this case, a method exists that can exactly minimize our particular energy function. If we imagine our energy function in parameter space to be an undulating field, whose lowest point we wish to find, we could say that this case is equivalent to being in the field on a clear, sunny day, seeing the lowest point and walking straight to it.

Now let us choose our probability model function, $f(x;\Theta)$, to be the sum of N normal distributions, so that $\Theta = \{\mu_{1,..,N}, \sigma_{1,..,N}\}$ and

$$f(x;\Theta) = \sum_{i=1}^{N} \mathcal{N}(x;\mu_i,\sigma_i) \tag{5}$$

This is equivalent to a sum-of-experts or mixture model, with equal weights on all the experts; having different weights is a trivial extension to the model. Again using the fact that a normal

distribution integrates to 1, we can see from Equation 2 that $\log Z(\Theta) = \log N$. However, now differentiating Equation 4 with respect to each of our model parameters produces equations dependent on other model parameters, so we cannot calculate the optimal model parameters straight off. Instead we can use the partial differential equations and a gradient descent method with line search to find a local minimum of energy in the parameter space.

Returning to our metaphor of the field, we could say that gradient descent with line search is equivalent to being in the field at night with a torch. We can either feel the gradient of the field at the point we're standing, or else estimate it by using the torch to see the relative height of the field a short distance in each direction from us (numerical differentiation using finite differences). Then, by shining the torch beam in our chosen direction of travel, it also allows us to see the lowest point in the field in that direction. We can then walk to that point, and iteratively choose a new direction and distance to walk.

Finally, let us choose our probability model function, $f(x;\Theta)$, to be the product of N normal distributions, so that

$$f(x;\Theta) = \prod_{i=1}^{N} \mathcal{N}(x;\mu_i,\sigma_i) \tag{6}$$

This is equivalent to a product-of-experts model. The partition function, $Z(\Theta)$, is now no longer a constant. We can see this by considering a model consisting of two normal distributions, both with $\sigma = 1$. If $\mu_1 = -\infty$ and $\mu_2 = \infty$ then $Z(\Theta) = 0$, while if $\mu_1 = \mu_2 = 0$ then $Z(\Theta) = 1/2\sqrt{\pi}$.

While it is possible, in this case, to compute the partition function exactly given Θ, let us imagine that the integration of Equation 2 is not algebraically tractable (as will be the case with other probability model functions). In this case we would need to use a numerical integration method to evaluate Equation 4, use finite differences to calculate the gradient at a given point in parameter space, and use a gradient descent method to find a local minimum. For high-dimensional data spaces the integration time is crippling, and a high-dimensional parameter space compounds this problem. This leads to a situation where we are trying to minimize an energy function that we cannot evaluate.

This is where CD helps us. Even though we cannot evaluate the energy function itself, CD provides a way to estimate the gradient of the energy function. If we return to our field metaphor, we now find ourselves in the field without any light whatsoever (i.e. we can't calculate energy), so we cannot establish the height of any point in the field relative to our own. CD effectively gives us a sense of balance, allowing us to the feel the gradient of the field under our feet. By taking very small steps in the direction of steepest gradient we can then find our way to a local minimum.

How does CD work?

As explained, CD estimates our energy function's gradient, given a set of model parameters, Θ, and our training data, $\mathbf{X}$. We will derive the gradient equation by firstly writing down the partial derivative of Equation 4:

$$\frac{\partial E(\mathbf{X};\Theta)}{\partial \Theta} = \frac{\partial \log Z(\Theta)}{\partial \Theta} - \frac{1}{K}\sum_{i=1}^{K} \frac{\partial \log f(x_i;\Theta)}{\partial \Theta} \tag{7}$$

$$= \frac{\partial \log Z(\Theta)}{\partial \Theta} - \left\langle \frac{\partial \log f(x;\Theta)}{\partial \Theta} \right\rangle_{\mathbf{X}} \tag{8}$$

where $\langle\cdot\rangle_{\mathbf{X}}$ is the expectation of $\cdot$ given the data distribution $\mathbf{X}$.

The first term on the right-hand side comes from the partition function, $Z(\Theta)$, which, as

Equation 2 shows, involves an integration over x. Substituting this in, we get

$$\begin{aligned}
\frac{\partial \log Z(\Theta)}{\partial \Theta} &= \frac{1}{Z(\Theta)} \frac{\partial Z(\Theta)}{\partial \Theta} && (9)\\
&= \frac{1}{Z(\Theta)} \frac{\partial}{\partial \Theta} \int f(x;\Theta)\ dx && (10)\\
&= \frac{1}{Z(\Theta)} \int \frac{\partial f(x;\Theta)}{\partial \Theta}\ dx && (11)\\
&= \frac{1}{Z(\Theta)} \int f(x;\Theta) \frac{\partial \log f(x;\Theta)}{\partial \Theta}\ dx && (12)\\
&= \int p(x;\Theta) \frac{\partial \log f(x;\Theta)}{\partial \Theta}\ dx && (13)\\
&= \left\langle \frac{\partial \log f(x;\Theta)}{\partial \Theta} \right\rangle_{p(x;\Theta)} && (14)
\end{aligned}$$

As discussed, this integration is generally algebraically intractable. However, in the form of Equation 14, it is clear that it can be numerically approximated by drawing samples from the proposed distribution, $p(x;\Theta)$.

Samples cannot be drawn directly from $p(x;\Theta)$ as we do not know the value of the partition function, but we can can use many cycles of Markov Chain Monte Carlo (MCMC) sampling to transform our training data (drawn from the target distribution) into data drawn from the proposed distribution. This is possible as the transformation only involves calculating the ratio of two probabilities, $p(x';\Theta)/p(x;\Theta)$, so the partition function cancels out. $\mathbf{X}^n$ represents the training data transformed using n cycles of MCMC, such that $\mathbf{X}^0 \equiv \mathbf{X}$. Putting this back into Equation 8, we get:

$$\frac{\partial E(\mathbf{X};\Theta)}{\partial \Theta} = \left\langle \frac{\partial \log f(x;\Theta)}{\partial \Theta} \right\rangle_{\mathbf{X}^\infty} - \left\langle \frac{\partial \log f(x;\Theta)}{\partial \Theta} \right\rangle_{\mathbf{X}^0} \tag{15}$$

We still have a computational hurdle to overcome—the many MCMC cycles required to compute an accurate gradient will take far too long. Hinton's assertion was that only a few MCMC cycles would be needed to calculate an approximate gradient. The intuition behind this is that after a few iterations the data will have moved from the target distribution (i.e. that of the training data) towards the proposed distribution, and so give an idea in which direction the proposed distribution should move to better model the training data. Empirically, Hinton has found that even 1 cycle of MCMC is sufficient for the algorithm to converge to the ML answer.

As such, bearing in mind that we wish to go downhill in order to minimize our energy function, our parameter update equation can be written as:

$$\Theta_{t+1} = \Theta_t + \eta \left(\left\langle \frac{\partial \log f(x;\Theta)}{\partial \Theta} \right\rangle_{\mathbf{X}^0} - \left\langle \frac{\partial \log f(x;\Theta)}{\partial \Theta} \right\rangle_{\mathbf{X}^1} \right) \tag{16}$$

where η is the step size factor, which should be chosen experimentally, based on convergence time and stability.