On Contrastive Divergence

Contrastive Divergence (CD) is an approximative Maximum Likelihood (ML) learning algorithm proposed by Geoffrey Hinton [2, 1].

1 Why CD?

We would like to model the probability of a data point x using a function of the form $f_\theta(x)$ where θ denotes a vector of model parameters. As we know, the probability of x, $p_\theta(x)$ must integrate to 1 over all x, therefore:

$$p_\theta(x) = \frac{1}{Z(\theta)} f_\theta(x) \tag{1}$$

where $Z(\theta)$, known as the normalizing constant or partition function, is defined as

$$Z(\theta) = \int f_\theta(x) dx. \tag{2}$$

Image that we are given a set of training data-points $X = \{x_1, \ldots, x_N\}$. We can learn our model parameters θ by maximizing the likelihood of a the training set X:

$$\mathcal{L}(\theta) = \prod_{i=1}^{N} \frac{1}{Z(\theta)} f_\theta(x_i). \tag{3}$$

or, equivalently, by minimizing the negative log of $\mathcal{L}(\theta)$, which we shall call energy:

$$\mathcal{E}_\theta(X) = \log Z(\theta) - \frac{1}{N} \sum_{i=1}^{N} \log f_\theta(x_i). \tag{4}$$

First, let us choose our probability model function $f_\theta(x)$ to be the pdf of a normal distribution $x \sim \mathcal{N}(\mu, \sigma)$ where $\theta = \{\mu, \sigma\}$:

$$f_\theta(x) = \mathcal{N}(x; \mu, \sigma). \tag{5}$$

By Kolmogorov's second axiom of probability, $\int f_\theta(x) dx = 1$, so that $\log Z(\theta) = 0$. Differentiation of (12) with respect to μ shows that the optimal μ is the mean of the training set X:

$$\frac{\partial}{\partial \mu} \mathcal{E}_\theta(X) = \frac{\partial}{\partial \mu} \log \int f_\theta(x) dx - \frac{1}{n} \sum_{i=1}^{N} \frac{\partial}{\partial \mu} \log f_\theta(x_i) \propto \frac{1}{2} \sum_{i=1}^{N} \frac{\partial}{\partial \mu} (x_i - \mu)^2,$$

$$\frac{\partial}{\partial \mu} \mathcal{E}_\theta(X)\big|_{\hat{\mu}} = 0 \implies \hat{\mu} = \frac{1}{N} \sum_{i=0}^{N} x_i.$$

Similarly, differentiation of (12) with respect to σ shows that the optimal σ is the square root of the variant of the training set X. Sometimes, as in this case, an optimization method exists that can exactly minimize our particular energy function. If we imagine our energy function to be an undulating landscape, whose lowest point we wish to find, then we could draw the metaphor of being in this field on a clear, sunny day, seeing the lowest point and walking straight to it.

Now, let us choose our probability model function $f_\theta(x)$ to be the sum of K normal distributions where $\theta = \{(\mu_k, \sigma_k) : k = 1, \ldots, K\}$:

$$f_\theta(x) = \sum_{k=1}^{K} \mathcal{N}(x; \mu_k, \sigma_k). \tag{6}$$

We usually refer to (6) as mixture or sum-of-experts model. Again, using the fact that a normal distribution must integrate to 1, we can inspect (2) to see that $\log Z(\theta) = \log N$ holds. However, differentiation of (12) with respect to our model parameters results in equations dependent on other model parameters. In this case, we cannot calculate the optimal parameters directly, i.e. in closed-form. Instead, we may use the partial differential equations and a gradient descent method with line search to find a local minimum of the energy in the parameter space.

Returning the metaphor of an undulating landscape, 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 where we are 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 (i.e. numerical differentiation using finite differences). Then, by shining the beam of the torch in our chosen direction of travel, it allows us to see the lowest point in the field in that direction. We can walk to that point, and iteratively choose a new direction and distance to walk.

Finally, let us choose our probability model function $f_\theta(x)$ to be the product of K normal distributions where $\theta = \{(\mu_k, \sigma_k) : k = 1, \ldots, K\}$:

$$f_\theta(x) = \prod_{k=1}^{K} \mathcal{N}(x; \mu_k, \sigma_k). \tag{7}$$

We usually refer to (7) as product-of-experts model. Note, the partition function $Z(\theta)$ is no longer a constant. Consider $K = 2$ normal distribution with fixed $\sigma = 1$. If $\mu_1 = -\infty$ and $\mu_2 = \infty$, then $Z(\theta) = 0$. If $\mu_1 = 0$ and $\mu_2 = 0$, then $Z(\theta) = \frac{1}{2}\sqrt{\pi}$. While, in this case, it is still possible to compute that partition function exactly given θ, let us imagine that the integration in (2) is not algebraically tractable. Then we would need numerical integration to evaluate (12), use finite difference to compute the gradient in parameter space, and use gradient descent to find a local energy minimum. For high-dimensional data-space computing the integral numerically is expensive, further, a high-dimensional

parameter-space compounds this problem. This leads to a situation where we are trying to minimize an energy function which we cannot evaluate.

This is where CD comes into play as a remedy. Even though we are not able to evaluate the energy function itself, CD provides a way to estimate the gradient of the energy function. Returning to our field metaphor one last time, we find ourselves in the field without any light whatsoever (i.e. we cannot calculate energy), so we cannot establish the height of any point in the field relative to our own. CD gives us a sense of balance which allows us to feel the gradient of the field under our feet. By taking very small steps in the direction of steepest gradient, we will eventually find our way to a local minimum.

2 How does CD work?

Consider a probability distribution over a vector x with model parameters θ

$$p_\theta(x) = \frac{1}{Z(\theta)} f_\theta(x) \tag{8}$$

where $Z(\theta)$, known as the normalizing constant or partition function is defined as

$$Z(\theta) = \int f_\theta(x) dx. \tag{9}$$

Maximum Likelihood (ML) learning of parameters θ given i.i.d. samples $X = \{x_1, \ldots, x_N\}$ may be done by gradient ascent:

$$\theta^{(\tau+1)} = \theta^{(\tau)} + \eta \frac{\partial \mathcal{L}(\theta)}{\partial \theta}\bigg|_{\theta^{(\tau)}} \tag{10}$$

where η denotes the learning rate and $\mathcal{L}(\theta)$ is known as the average log-likelihood

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x_i). \tag{11}$$

We want to maximize the likelihood, or equivalently, minimizing the negative of $\mathcal{L}(\theta)$, which we shall call energy:

$$\mathcal{E}_\theta(X) = \log Z(\theta) - \frac{1}{N} \sum_{i=1}^{N} \log f_\theta(x_i). \tag{12}$$

Since we want to maximize the likelihood, we will derive the gradient equation by firstly writing down the partial derivative of (12):

$$\frac{\partial \mathcal{E}_\theta(X)}{\partial \theta} = \frac{\partial \log Z(\theta)}{\partial \theta} - \frac{1}{N}\sum_{i=1}^{N} \frac{\partial \log f_\theta(x_i)}{\partial \theta} \tag{13}$$

$$= \frac{\partial \log Z(\theta)}{\partial \theta} - \left\langle \frac{\partial \log f_\theta(x)}{\partial \theta} \right\rangle_X \tag{14}$$

where $\langle \cdot \rangle_X$ is the expectation with respect to the data distribution of X. The first term on the right-hand side comes from the partition function $Z(\theta)$, which, as equation (2), involves the integration over x. By substitution, we get

$$\frac{\partial \log Z(\theta)}{\partial \theta} = \frac{1}{Z(\theta)} \frac{\partial Z(\theta)}{\partial \theta} \tag{15}$$

$$= \frac{1}{Z(\theta)} \frac{\partial}{\partial \theta} \int f_\theta(x) dx \tag{16}$$

$$= \frac{1}{Z(\theta)} \int \frac{\partial f_\theta(x)}{\partial \theta} dx \tag{17}$$

$$= \frac{1}{Z(\theta)} \int f_\theta(x) \frac{\partial \log f_\theta(x)}{\partial \theta} dx \tag{18}$$

$$= \int p_\theta(x) \frac{\partial \log f_\theta(x)}{\partial \theta} dx \tag{19}$$

$$= \left\langle \frac{\partial \log f_\theta(x)}{\partial \theta} \right\rangle_{p_\theta(x)}. \tag{20}$$

This integration is algebraically intractable. However, in the form of (20), we can see that the expectation can be numerically approximated by drawing samples from the proposed distribution $p_\theta(x)$. However, samples cannot be drawn directly from $p_\theta(x)$ as we can not evaluate the partition function, but we may 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, namely the quotient $p_\theta(x')/p_\theta(x)$, so the partition function cancels out.

Let X^n denote the transformed training data using n cycles of MCMC, such that $X^0 = X$. Putting this back into (20), we get:

$$\frac{\partial \mathcal{E}_\theta(X)}{\partial \theta} = \left\langle \frac{\partial \log f_\theta(x)}{\partial \theta} \right\rangle_{X^\infty} - \left\langle \frac{\partial \log f_\theta(x)}{\partial \theta} \right\rangle_{X^0} \tag{21}$$

where $\langle \cdot \rangle_{X^\infty}$ denotes the expectation with respect to the model distribution $p_\infty(x) = p_\theta(x)$ and $\langle \cdot \rangle_{X^0}$ denotes the expectation with respect to the data distribution $p_0(x) = \frac{1}{N}\sum_{i=1}^{N} \delta(x - x_i)$.

Lastly, we still have the computational hurdle of too many MCMC cycles which are required to compute an accurate estimate of the gradient. Hinton assumed that only very few MCMC cycles would be needed to calculate an approximate gradient.

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:

$$\theta^{(\tau+1)} = \theta^{(\tau)} - \eta \left(\left\langle \frac{\partial \log f_\theta(x)}{\partial \theta} \right\rangle_{X^1} - \left\langle \frac{\partial \log f_\theta(x)}{\partial \theta} \right\rangle_{X^0} \right) \tag{22}$$

where η is the step size.

3 Why the name?

Maximum likelihood learning minimizes the Kullback-Leibler divergence:

$$KL\left(p_0 \| p_\infty\right) = \int p_0(x) \log \frac{p_0(x)}{p_\theta(x)}. \tag{23}$$

Contrastive divergence learning approximately follows the gradient of the difference of two divergences:

$$CD_n = KL\left(p_0 \| p_\infty\right) - KL\left(p_n \| p_\infty\right). \tag{24}$$

In CD learning, we start the Markov chain at the data distribution p_0 and run the chain for a small number of n steps (e.g. $n = 1$). This greatly reduces both the computation per gradient step and the variance of the estimated gradient.

References

[1] Miguel A. Carreira-Perpinan and Geoffrey E. Hinton. On contrastive divergence learning. 2005.

[2] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, August 2002.