Paper Review: Training Products of Experts by Minimizing Contrastive Divergence

Beomsu Kim*

June 28, 2021

Paper Information.

- Geoffrey E. Hinton. Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation, 14(8):1771–1800, 2002.
- 5000+ citations.
- 500+ citations per year since 2015.

1 Introduction

- One way of modeling a complicated, high dimensional data distribution is to combine a large number of relatively simple probabilistic models.
- Gaussian Mixture Model with EM or gradient ascent.

$$p(\mathbf{d} \mid \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{d} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \qquad \sum_{k=1}^K \pi_k = 1, \qquad \pi_k \geq 0.$$

- This paper proposes the Product of Experts (PoE) model. Multiplying simple models and renormalizing

$$p(\mathbf{d} \mid \{\theta_k\}_{k=1}^K) = \frac{\prod_k p_k(\mathbf{d} \mid \theta_k)}{\sum_{\mathbf{c}} \prod_k p_k(\mathbf{c} \mid \theta_k)}$$

  where $\mathbf{c}$ indexes all possible vectors in the data space. For continuous spaces, the sum is replaced by an integral.
- Each expert $p_k(\mathbf{d} \mid \theta_k)$ can constrain a different subset of the dimension in a high-dimensional space and their product will then constraint all of the dimensions.
- For modeling handwritten digits, one low-resolution model can generate images that have the approximate overall shape of the digit and other more local models can ensure that small image patches contain segments of stroke with the correct fine structure.
- On the other hand, for mixture models, each model must know how to produce an entire digit.
- Applications from training Boltzmann Machines to training deep energy based models.

*Department of Mathematical Sciences, KAIST. Email beomsu.kim@kaist.ac.kr

2 Learning PoE by Maximizing Likelihood

- Recall that we defined PoE by

$$p(\mathbf{d} \mid \{\theta_k\}_{k=1}^K) = \frac{\prod_k p_k(\mathbf{d} \mid \theta_k)}{\sum_{\mathbf{c}} \prod_k p_k(\mathbf{c} \mid \theta_k)} \tag{1}$$

  where $\mathbf{c}$ indexes all possible vectors in the data space.

- The obvious way to fit a PoE to a set of observed i.i.d. vectors is to perform gradient ascent on the log likelihood of each observed vector, $\mathbf{d}$, under the PoE. This is given by

$$\begin{aligned}
\frac{\partial}{\partial \theta_n} \log p(\mathbf{d} \mid \{\theta_k\}_{k=1}^K) &= \frac{\partial}{\partial \theta_n} \log \left(\frac{\prod_k p_k(\mathbf{d} \mid \theta_k)}{\sum_{\mathbf{c}} \prod_k p_k(\mathbf{c} \mid \theta_k)} \right) \\
&= \frac{\partial}{\partial \theta_n} \sum_k \log p_k(\mathbf{d} \mid \theta_k) - \frac{\partial}{\partial \theta_n} \log \left(\sum_{\mathbf{c}} \prod_k p_k(\mathbf{c} \mid \theta_k) \right) \\
&= \frac{\partial}{\partial \theta_n} \log p_n(\mathbf{d} \mid \theta_n) - \sum_{\mathbf{c}} \frac{\prod_{k \neq n} p_k(\mathbf{c} \mid \theta_k)}{\sum_{\mathbf{c}} \prod_k p_k(\mathbf{c} \mid \theta_k)} \cdot \frac{\partial}{\partial \theta_n} p_n(\mathbf{c} \mid \theta_n) \\
&= \frac{\partial}{\partial \theta_n} \log p_n(\mathbf{d} \mid \theta_n) - \sum_{\mathbf{c}} \frac{\prod_k p_k(\mathbf{c} \mid \theta_k)}{\sum_{\mathbf{c}} \prod_k p_k(\mathbf{c} \mid \theta_k)} \cdot \frac{1}{p_n(\mathbf{c} \mid \theta_n)} \cdot \frac{\partial}{\partial \theta_n} p_n(\mathbf{c} \mid \theta_n) \\
&= \frac{\partial}{\partial \theta_n} \log p_n(\mathbf{d} \mid \theta_n) - \sum_{\mathbf{c}} \frac{\prod_k p_k(\mathbf{c} \mid \theta_k)}{\sum_{\mathbf{c}} \prod_k p_k(\mathbf{c} \mid \theta_k)} \cdot \frac{\partial}{\partial \theta_n} \log p_n(\mathbf{c} \mid \theta_n) \\
&= \frac{\partial}{\partial \theta_n} \log p_n(\mathbf{d} \mid \theta_n) - \sum_{\mathbf{c}} p(\mathbf{c} \mid \{\theta_k\}_{k=1}^K) \cdot \frac{\partial}{\partial \theta_n} \log p_n(\mathbf{c} \mid \theta_n).
\end{aligned} \tag{2}$$

- The second term on the RHS of Equation (2) is the expected derivative of the log probability of an expert on fantasy data, $\mathbf{c}$, that is generated from the PoE.
- So, assuming each of the individual experts has a tractable derivative, the obvious difficulty in estimating the derivative of the log probability of the data under PoE is generating correctly distributed fantasy data.
- It is possible to use rejection sampling: each expert generates a data vector independently and this process is repeated until all the experts happen to agree. However, this is typically very inefficient.
- A MCMC method that uses Gibbs sampling is typically much more efficient.
- Samples from the PoE distribution have very high variance since they come from all over the model's distribution. This makes gradient ascent difficult.

A brief explanation on sampling with tractable pdf.

- Suppose we have a one-dimensional pdf $p(d)$.
- We can then calculate its cdf $F(d)$ by integrating $p(d)$.
- Let U be a uniform random variable on $[0, 1]$.
- What distribution does $X = F^{-1}(U)$ follow? If F_X is the cdf of X,

  $$F_X(d) = \mathbb{P}(X \leq d) = \mathbb{P}(F^{-1}(U) \leq d) = \mathbb{P}(U \leq F(d)) = F(d)$$

  so we see that $F^{-1}(U)$ has the distribution $p(d)$.
- By passing samples from the uniform distribution on $[0, 1]$ through F^{-1}, we get samples from $p(d)$.

A brief explanation on MCMC.

- Suppose we want to sample from an unnormalized or intractable distribution $p(\mathbf{d})$. That is,

  $$\int_{\mathbf{c}} p(\mathbf{c}) \neq 1 \quad \text{or} \quad F(\mathbf{d}) \text{ is difficult to compute.}$$

  where $\mathbf{c}$ indexes all possible vectors in the data space. In this paper, $p(\mathbf{d}) = \prod_k p_k(\mathbf{d} \mid \theta_k)$.
- If the space is extremely large, the integral is intractable.
- Then, we cannot calculate its cdf, so we cannot apply the inverse transform sampling technique.
- Denote an arbitrary initial distribution by $Q_0(\mathbf{d})$.
- Define a transition kernel K which is a function that inputs and outputs distributions.

  $$Q^1 = KQ^0, \qquad Q^n = KQ^{n-1} = K^n Q_0.$$

- K is typically a function of $p(\mathbf{d})$.
- For an appropriate choice of K,

  $$Q^\infty(\mathbf{d}) = \lim_{n\to\infty} Q^n(\mathbf{d}) = \frac{p(\mathbf{d})}{\int_{\mathbf{c}} p(\mathbf{c})}.$$

- What property does Q^∞ have?

  $$Q^n = KQ^{n-1} \implies Q^\infty = \lim_{n\to\infty} Q^n = \lim_{n\to\infty} KQ^{n-1} = KQ^\infty.$$

  That is, Q^∞ is an eigenvector of K with eigenvalue 1. So

  $$Q^n = Q^{n-1} \implies KQ^{n-1} = Q^{n-1} \implies Q^{n-1} = Q^\infty \implies Q^{n-1}(\mathbf{d}) = \frac{p(\mathbf{d})}{\int_{\mathbf{c}} p(\mathbf{c})}.$$

  We also say Q^∞ is a *fixed point* of K.
- MCMC is an iterative sampling technique, which
  - at the 0-th step, samples from Q^0,
  - at the n-th step, uses samples from Q^{n-1} to sample from Q^n.
- Gibbs sampling is a type of MCMC sampling with coordinate-wise updates.
- For more information, see
  - *An Introduction to MCMC for Machine Learning*,
  - *Gibbs Sampling for the Uninitiated*.

3 Learning by Minimizing Contrastive Divergence

- Let Q^0 be the data distribution and Q^∞ be the equilibrium distribution Q^∞ that is produced by prolonged Gibbs sampling from the generated model.

$$Q^\infty(\mathbf{d}) = p(\mathbf{d} \mid \{\theta_k\}_{k=1}^K).$$

- Maximizing the log likelihood of the data (averaged over the data distribution) is equivalent to minimizing the Kullback-Leibler divergence between Q^0 and Q^∞.

$$D_{\mathrm{KL}}(Q^0 \| Q^\infty) = \sum_{\mathbf{d}} Q^0(\mathbf{d}) \log Q^0(\mathbf{d}) - \sum_{\mathbf{d}} Q^0(\mathbf{d}) \log Q^\infty(\mathbf{d}) = -H(Q^0) - \mathbb{E}_{Q^0}[\log Q^\infty(\mathbf{d})] \quad (3)$$

  Since $H(Q^0)$ does not depend on the parameters of the model,

$$\arg\min_{\theta_k} D_{\mathrm{KL}}(Q^0 \| Q^\infty) = \arg\max_{\theta_k} \mathbb{E}_{Q^0}[\log Q^\infty(\mathbf{d})].$$

- Recall Equation (2)

$$\begin{aligned} \frac{\partial \log p(\mathbf{d} \mid \{\theta_k\}_{k=1}^K)}{\partial \theta_n} &= \frac{\partial \log p_n(\mathbf{d} \mid \theta_n)}{\partial \theta_n} - \sum_{\mathbf{c}} p(\mathbf{c} \mid \{\theta_k\}_{k=1}^K) \cdot \frac{\partial \log p_n(\mathbf{c} \mid \theta_n)}{\partial \theta_n} \\ &= \frac{\partial \log p_n(\mathbf{d} \mid \theta_n)}{\partial \theta_n} - \mathbb{E}_{Q^\infty}\left[\frac{\partial \log p_n(\mathbf{c} \mid \theta_n)}{\partial \theta_n}\right] \end{aligned}$$

- Equation (2), averaged over the data distribution, can be rewritten as

$$\mathbb{E}_{Q^0}\left[\frac{\partial \log Q^\infty(\mathbf{d})}{\partial \theta_n}\right] = \mathbb{E}_{Q^0}\left[\frac{\partial \log p_n(\mathbf{d} \mid \theta_n)}{\partial \theta_n}\right] - \mathbb{E}_{Q^\infty}\left[\frac{\partial \log p_n(\mathbf{c} \mid \theta_n)}{\partial \theta_n}\right]. \quad (4)$$

- To perform a single step of gradient ascent, we need to run Gibbs sampling for a long time!
- There is a simple and effective alternative to maximum likelihood learning which eliminates almost of all of the computation required to get samples from Q^∞ and also eliminates much of the variance that masks the gradient signal.
- Instead of minimizing $D_{\mathrm{KL}}(Q^0 \| Q^\infty)$, we minimize the *contrastive divergence*

$$D_{\mathrm{KL}}(Q^0 \| Q^\infty) - D_{\mathrm{KL}}(Q^1 \| Q^\infty)$$

  where Q^1 is the distribution over one step of Gibbs sampling.

- Because Q^1 is one step closer to Q^∞ than Q^0, we are guaranteed that

$$D_{\mathrm{KL}}(Q^0 \| Q^\infty) \geq D_{\mathrm{KL}}(Q^1 \| Q^\infty) \implies D_{\mathrm{KL}}(Q^0 \| Q^\infty) - D_{\mathrm{KL}}(Q^1 \| Q^\infty) \geq 0.$$

- What property does the solution have?

$$Q^0 = Q^1 \implies Q^0 = Q^\infty \implies \text{PoE perfectly models the data distribution.}$$

- How do we optimize the contrastive divergence objective? Observe that by (3) and (4),

$$\begin{aligned}
-\frac{\partial}{\partial \theta_n} D_{\mathrm{KL}}(Q^0 \| Q^\infty) &= \frac{\partial}{\partial \theta_n}(H(Q^0) + \mathbb{E}_{Q^0}[\log Q^\infty(\mathbf{d})]) \\
&= \mathbb{E}_{Q^0}\left[\frac{\partial \log Q^\infty(\mathbf{d})}{\partial \theta_n}\right] \\
&= \mathbb{E}_{Q^0}\left[\frac{\partial \log p_n(\mathbf{d} \mid \theta_n)}{\partial \theta_n}\right] - \mathbb{E}_{Q^\infty}\left[\frac{\partial \log p_n(\mathbf{c} \mid \theta_n)}{\partial \theta_n}\right].
\end{aligned}$$

  We also have (recall that $Q^1 = KQ^0$, where the transition kernel K depends on $\prod_k p_k(\mathbf{d} \mid \theta_k)$)

$$\begin{aligned}
-\frac{\partial}{\partial \theta_n} D_{\mathrm{KL}}(Q^1 \| Q^\infty) &= -\frac{\partial Q^1}{\partial \theta_n}\frac{\partial D_{\mathrm{KL}}(Q^1 \| Q^\infty)}{\partial Q^1} - \frac{\partial Q^\infty}{\partial \theta_n}\frac{\partial D_{\mathrm{KL}}(Q^1 \| Q^\infty)}{\partial Q^\infty} \\
&= -\frac{\partial Q^1}{\partial \theta_n}\frac{\partial D_{\mathrm{KL}}(Q^1 \| Q^\infty)}{\partial Q^1} + \mathbb{E}_{Q^1}\left[\frac{\partial \log p_n(\hat{\mathbf{d}} \mid \theta_n)}{\partial \theta_n}\right] - \mathbb{E}_{Q^\infty}\left[\frac{\partial \log p_n(\mathbf{c} \mid \theta_n)}{\partial \theta_n}\right]
\end{aligned}$$

  where $\hat{\mathbf{d}} \sim Q^1(\hat{\mathbf{d}})$. It follows that

$$\begin{aligned}
\frac{\partial}{\partial \theta_n}(D_{\mathrm{KL}}(Q^0 \| Q^\infty) - D_{\mathrm{KL}}(Q^1 \| Q^\infty)) = \mathbb{E}_{Q^0}\left[\frac{\partial \log p_n(\mathbf{d} \mid \theta_n)}{\partial \theta_n}\right] - \mathbb{E}_{Q^1}\left[\frac{\partial \log p_n(\hat{\mathbf{d}} \mid \theta_n)}{\partial \theta_n}\right] \\
+ \frac{\partial Q^1}{\partial \theta_n}\frac{\partial D_{\mathrm{KL}}(Q^1 \| Q^\infty)}{\partial Q^1}. \qquad (5)
\end{aligned}$$

- If each expert is chosen to be tractable, it is possible to compute the exact values of the derivatives of $\log p_m(\mathbf{d} \mid \theta_m)$ and $\log p_m(\hat{\mathbf{d}} \mid \theta_m)$.
- It is also straightforward to sample from Q^0 and Q^1. To sample from Q^0, just randomly pick samples in the training set. To sample from Q^1, simply run one step of Gibbs sampling.
- The third term on the RHS of (5) is problematic, but extensive simulations show that it can safely be ignored because it is small and it seldom opposes the resultant of the other two terms.
- The parameters of the experts can therefore be adjusted in proportion to the approximate derivative of the contrastive divergence.

$$\Delta\theta_m \propto \mathbb{E}_{Q^0}\left[\frac{\partial \log p_n(\mathbf{d} \mid \theta_n)}{\partial \theta_n}\right] - \mathbb{E}_{Q^1}\left[\frac{\partial \log p_n(\hat{\mathbf{d}} \mid \theta_n)}{\partial \theta_n}\right]$$

- The difference in the derivatives of the data vectors and their reconstructions have some variance because the reconstruction procedure is stochastic. But when the PoE is modeling the data moderately well, the one-step reconstructions will be very similar to the data so the variance will be very small.

7 PoEs and Boltzmann Machines

- Consider a training set of binary vectors which we will assume are binary images.
- The training set can be modeled using a two-layer network called a RBM in which stochastic, binary pixels are connected to stochastic, binary feature detectors.
- The pixels correspond to visible units of the RBM because their states are observed.
- The feature detectors correspond to hidden units.
- A joint configuration, $(\mathbf{v}, \mathbf{h})$ of the visible and hidden units has an energy given by

$$E(\mathbf{v}, \mathbf{h}, \{\mathbf{w}_k\}_{k=1}^K) = -\sum_{k=1}^K \langle \mathbf{w}_k, \mathbf{v} \rangle \cdot h_k$$

  for weights $\mathbf{w}_k$, $k = 1, \ldots, K$.
- General RBMs also have biases, which I have omitted for simplicity.
- RBM defines the probability distribution

$$p(\mathbf{v}, \mathbf{h} \mid \{\mathbf{w}_k\}_{k=1}^K) = \frac{1}{Z} e^{-E(\mathbf{v}, \mathbf{h}, \{\mathbf{w}_k\}_{k=1}^K)}, \qquad Z = \sum_{\mathbf{v}} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h}, \{\mathbf{w}_k\}_{k=1}^K)}.$$

- How are RBMs related to PoEs? Observe that

$$\begin{aligned} p(\mathbf{v} \mid \{\mathbf{w}_k\}_{k=1}^K) &= \sum_{\mathbf{h} \in \{0,1\}^K} p(\mathbf{v}, \mathbf{h} \mid \{\mathbf{w}_k\}_{k=1}^K) \\ &= \frac{1}{Z} \sum_{\mathbf{h} \in \{0,1\}^K} \exp\left\{ -\sum_{k=1}^K \langle \mathbf{w}_k, \mathbf{v} \rangle \cdot h_k \right\} \\ &= \frac{1}{Z} \sum_{\mathbf{h} \in \{0,1\}^K} \prod_{k=1}^K \exp\left\{ -\langle \mathbf{w}_k \cdot \mathbf{v} \rangle \cdot h_k \right\} \\ &= \frac{1}{Z} \prod_{k=1}^K (1 + \exp\left\{ -\langle \mathbf{w}_k \cdot \mathbf{v} \rangle \right\}) \qquad \text{(why?)} \end{aligned}$$

  so if we define the "expert"

$$p_k(\mathbf{v} \mid \mathbf{w}_k) = 1 + \exp\left\{ -\langle \mathbf{w}_k \cdot \mathbf{v} \rangle \right\},$$

  we get the PoE model

$$p(\mathbf{v} \mid \{\mathbf{w}_k\}_{k=1}^K) = \frac{1}{Z} \prod_{k=1}^K p_k(\mathbf{v} \mid \mathbf{w}_k).$$

- Hence, we can train RBMs via contrastive divergence.
- For more information on RBMs, see
  - *A Practical Guide to Training RBMs.*