E0-270 Machine Learning
Report of the Term Project

Energy Based Models - Improved Contrastive Divergene Training

Shingala Jaydeep Jaysukhbhai (SR: 21076)
Department of Computer Science and Automation, IISc, Bangalore.
jaydeeps@iisc.ac.in

Abstract

In recent years Energy Based Models have recived good attentions from perspective of Generative Machine learning. EBMs under some conditions have been proved to beat GANs for image synthesis, generation and super-resolution tasks. EBMs are used for learning underlying data distributions. I have presented different training techniques of EBMs. Further I have explored Contrastive and Improved Contrastive divergence training algorithms of EBMs. I have explored learning algorithms, pros and cons, and demonstrated experimental comparisons with contrastive divergence learning for sample code. Further I explored another improvements that can be applied to training method for batter performance like Gibbs-sampling, data augmentation and persistance contrastive divergence. The construction and training of Restricted Boltzman Machines is also shown with code. The main paper that i explored is Improved contrastive divergence training of Energy Based Models (1)

1 Energy Based Models

1.1 Introduction

Energy-based models (EBMs) are a class of probabilistic models used in machine learning and artificial intelligence. These models define a probability distribution over the data by defining an energy function that assigns a scalar value (often referred to as energy) to each possible configuration of the data. The energy function is designed such that low-energy configurations correspond to high-probability states, and high-energy configurations correspond to low-probability states. The goal of training an EBM is to learn the parameters of the energy function so that it assigns low energy to configurations that are likely to occur in the data and high energy to configurations that are unlikely to occur. This is kind of given some model parameters say θ, the goal is to maximise the likelihood and by that trying to learn underlying distribution for generative tasks. So, the energy will be,

$$E_\theta(x) = -\log(p(x|\theta))$$

and the probability distribution will be,

$$q_\theta(x) = \frac{1}{Z(\theta)} \exp(-E_\theta(x)) \text{ where } Z(\theta) = \int e^{-E_\theta(x)} dx \text{ if } x \in \text{contineous space}$$

here, calculating normalising constant Z is intractable as it is integration over entire feature space, we need some sampling techniques to approximate the quantity.

1.2 Different training algorithms for EBMs

There are several training algorithms for EBMs, each with their own strengths and weaknesses. Here are some of the most commonly used training algorithms for EBMs:

1. Maximum likelihood estimation (MLE): This is a standard method for training probabilistic models, including EBMs. The goal is to maximize the log-likelihood of the training data with respect to the model parameters using gradient-based optimization techniques. MLE is conceptually simple and easy to implement, but it can suffer from slow convergence and overfitting.

The objective function for MLE in EBMs can be expressed as:

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[E(x^{(i)}, \theta) - \log Z(\theta) \right]$$

where θ are the parameters of the model, $x^{(i)}$ are the input data samples, n is the number of training samples, $E(x^{(i)}, \theta)$ is the energy of the model for a given input sample $x^{(i)}$, and $\log Z(\theta)$ is the logarithm of the partition function of the model. The goal of MLE is to find the parameters w that maximize the likelihood of the training data, which is equivalent to minimizing the negative log-likelihood $-L(\theta)$.

2. Contrastive divergence (CD): This is an approximation to the score matching estimator that uses a Markov chain to estimate the expected feature activations under the model distribution. The gradient is then estimated using the difference between the expected feature activations under the data distribution and the model distribution. CD is computationally efficient and can work well for small and medium-sized models, but it can be biased and have high variance for high-dimensional models.

3. Persistent contrastive divergence (PCD): This is a variant of CD that uses a persistent Markov chain that is updated at each step using the transition probabilities defined by the EBM. The persistent chain is initialized with a training example and is kept persistent across iterations, so that the same chain is used to estimate the gradient at each step. PCD can reduce the bias introduced by CD and improve the convergence rate, especially for high-dimensional models.

4. Wake-sleep algorithm: This is a variant of MLE that uses a two-phase training procedure to estimate the gradients. In the "wake" phase, the model is trained to generate samples that match the data distribution, and in the "sleep" phase, the model is trained to generate samples that match a specified prior distribution. The gradient is then estimated using the difference between the expected feature activations under the data distribution and the prior distribution. The wake-sleep algorithm can be more robust to overfitting and can work well for models with complex dependencies, but it can be computationally expensive and difficult to tune.

Overall, the choice of training algorithm depends on the specific characteristics of the model and the training data, as well as the computational resources available.

1.3 Applications

Energy-Based Models have received an good amount interest. EBMs are often used in unsupervised learning tasks, such as generative modeling and density estimation, as well as in supervised learning tasks, such as classification. They have been applied to a wide range of applications, including computer vision, natural language processing, and speech recognition, among others. In recent and have been applied to realistic image generation, 3D shapes synthesis, out of distribution and adversarial robustness, compositional generation, memory modeling, text generation, video generation, reinforcement learning, continual training, protain design and folding and biologically-plausible training.

2 Training of EBMs using contrastive Divergence Algorithm

Constructive divergence is a learning algorithm for training energy-based models (EBMs). The basic idea behind CD is to approximate the gradient of the log-likelihood of the data with respect to the model parameters, which is difficult to compute exactly, using a simpler and faster procedure.
Basically we want to find the probability distribution over training dataset X that satisfies following 2 conditions:

1. The probability distribution needs to assign any possible value of X a non-negative value: $P(X) >= 0$.

2. The probability density must sum/integrate to 1 over all possible inputs: $\int_x f(X)dx = 1$.

2.1 Simple Contrastive Divergence Algorithm

The fundamental idea is that we can turn any function that predicts values larger than zero into a probability distribution by dividing by its volume. we have a neural network, which has as output a single neuron, like in regression. We can call this network $E\theta$(x), where θ are our parameters of the network, and the input data x(e.g. an image). The output of is a scalar value between $+\infty$ and $-\infty$. Now, we can use basic probability theory to normalize the scores of all possible inputs.
In this form, the energy function can be thought of as a measure of the compatibility of the input configuration with the parameters of the model. A low energy value indicates a good fit between the input configuration and the model parameters, while a high energy value indicates a poor fit. The goal of training the energy-based model is to learn the optimal parameters that result in the lowest energy value for a given input configuration.
Energy-Based Models represent the likelihood of a probability distribution $p_D(x)$ for $x \in \mathbb{R}^D$ as $p_\theta(x) = \frac{1}{Z(\theta)} \exp(-E_\theta(x))$, where the function $E_\theta(x) : \mathbb{R}^D \to \mathbb{R}$ is known as the energy function, and $Z(\theta) = \int_x \exp(-E_\theta(x))$ is known as the partition function. Thus, an EBM can be represented by a neural network that takes x as input and outputs a scalar.
Now Calculating $Z(\theta)$ is intractable or in terms of exponential form. but for extact likelihood we need $Z(\theta)$. so we try to approximate $Z(\theta)$ in contrastive divergence using Markov Chain Monte Carlo(MCMC) and Gibbs sampling.
The gradient of log-likelihood with respect to a data sample x can be represented as

$$\frac{\partial \log p_\theta(x)}{\partial \theta} = -\left(\frac{\partial E_\theta(x)}{\partial \theta} - \mathbb{E}_{p_\theta(x')}\left[\frac{\partial E_\theta(x')}{\partial \theta}\right]\right).$$

Above equestion is still intractable as it requires us to run MCMC to find samples from the model distribution $p_D(x)$. MCMC have two phases which are burn-in phase and sampling phase. As sometimes in MCMC burn-in phase and mixing can take exponential time, this sampling is still possible but intractable.
So, in contrastive divergence we propose objective function as:

$$KL(p_D(x)||p_\theta(x)) - KL\left(\Pi_t\theta(p_D(x))||p_\theta(x)\right)$$

where Π_θ represents a MCMC transition kernel for p_θ, and $\Pi_t\theta(p_D(x))$ represents t sequential MCMC transitions starting from $p(x)$.
The objective function consists of two terms, both of which are KL (Kullback-Leibler) divergences. The KL divergence is a measure of how different two probability distributions are from each other.
The first term is the KL divergence between the data distribution and the EBM distribution. This term measures how well the EBM is able to model the data. The objective function seeks to minimize this term, which means that the EBM is trying to become as similar to the data distribution as possible.
The second term is the KL divergence between the EBM distribution and a distribution obtained by running a finite number of Markov Chain Monte Carlo (MCMC) steps on the data distribution. MCMC is a sampling technique used to approximate a probability distribution. This term measures how well the EBM can generate samples that are similar to the data. The objective function seeks to minimize this term as well, which means that the EBM is trying to generate samples that are as similar to the data as possible.
So, the objective function is trying to find the EBM that best models the data and generates samples that are similar to the data. By minimizing the difference between the two KL divergences, the EBM is encouraged to become as close to the data distribution as possible, while still being able to generate samples that are similar to the data.

2.2 Improved Contrastive Divergence Algorithm

When taking the negative gradient of the contrastive divergence objective, we get

$$-\mathbb{E}_{p_D(x)}\left[\frac{\partial E_\theta(x)}{\partial \theta}\right] + \mathbb{E}_{q_\theta(x')}\left[\frac{\partial E_\theta(x')}{\partial \theta}\right] - \frac{\partial q(x')}{\partial \theta}\frac{\partial KL(q_\theta(x')||p_\theta(x'))}{\partial q_\theta(x')} \tag{1}$$

Now, in simple contrastive divergence Hinton (3) was ignoring the last term in equation 1, stating that term is negligible and small compared with previous two terms and hence we can ignore that for computational simplicity. But Here, authors (1) of this paper showed empirically and with practical implementations that we should not ignore that term. if we do not ignore that term then we can get high quality and faster convergence.
Authors (1) also showed that, this KL-term can be approximated by different strategies. So, we construct a new joint loss expression L_{Full}, consisting of traditional contrastive loss L_{CD} and a new loss expression L_{KL}, to accurately exhibit all three gradient terms. Specifically, we have $L_{Full} = L_{CD} + L_{KL}$ where L_{CD} is,

$$L_{CD} = E_{p_D(x)}[E_\theta(x)] - E_{stop\,grad(q_\theta(x'))}[E_\theta(x')] \tag{2}$$

$$L_{KL} = E_{q_\theta(x)}[E_{stop\,grad(\theta)}(x)] + E_{q_\theta(x)}[\log(q_\theta(x))] \tag{3}$$

Though estimating L_{KL} is hard, we need it for both speeding up and stabilizing training of EBMs. The Equation 2 encourages the energy function to assign low energy to real samples (implying high likelihood) and high energy to generated samples (implying lower likelihood). Now, if we simply only optimise equation 2 then that leads to narrow energy landscape that makes sampling difficult. The KL term counters it as this encourages sampling to closely approximate the underlying distribution by allowing samples from lower energy as well as different diversity.

3 Estimation of the Missing gradient KL term

A difficulty when training EBMs is that underlying MCMC chains fail to mix and cover the EBM distribution. To enable more effective mixing of MCMC chains, we intersperse Langevin sampling with data augmentation transitions. This enables image sampling chains from our model to travel across large number of modes in the energy landscape.
There are some technies that can be used are:

3.1 Langevin sampling

This is a type of transition that involves adding a noise term to the current sample and then taking a gradient step towards the mode of the EBM distribution. Langevin sampling helps the chains explore the energy landscape of the EBM more effectively, by pushing the chains towards areas of high probability density. For batter performance and improved generation qualities, authors have used langevian sampling extensively.

3.2 Entropy Estimation

The goal of maximizing entropy is to encourage the model to generate diverse and novel samples that cover the entire support of the data distribution. The estimator is used to estimate the entropy of a distribution $p(x)$ by sampling n different points from $p(x)$. The entropy $H(p_\theta(x))$ is estimated using the formula $H(p_\theta(x)) = \frac{1}{n}\sum_{i=1}^{n} \ln(n * NN(x_i, X)) + O(1)$, where $NN(x_i, X)$ denotes the nearest neighbor distance of x_i to any other data point in X. The estimator has been shown to be mean square consistent with a root-n convergence rate.

4 Other proposed improvements

There are some other improvements to the training algorithm suggested are discussed. These algorithms include

4.1 Data augmentation transitions:

Sometimes our MCMC sampling chain may fall into trap where it converges to same sample again and again. This happens because of in real time big difference between similar qualitative images but with our finite number of steps we can not reach/ overcome that difference. Though the proposed KL term regularises diversity but alone it is not enough for good samples. Thus Data augmentation involves adding a random perturbation to the input data, and then running the EBM forward to generate a new sample. This helps the chains explore different modes of the EBM distribution, by creating new "augmented" versions of the data that the EBM has not seen before.
Data augmentation transitions are an important tool for training EBMs using MCMC, as they allow the chain to explore a wider range of modes in the EBM distribution. By introducing new augmented data points into the chain, data augmentation transitions help to improve the mixing of the chain and generate high-quality samples from the EBM distribution. To be specific we can use augmentation techniques like different combination of color, horizontal flip, rescaling, and guassian blur augmentations. During training time we initialise MCMC sampling from a data augmentation applied to an input sampled from the buffer of past samples. at test time when we are in the process of generating samples we apply a random augmentation to the input after every 20 or so steps of lengevian sampling.

4.2 Compositional Multi-scale Generation

The approach involves breaking down the input image into different resolutions or scales (full, half, and quarter resolution) and applying energy functions to each of these scales separately. By using multiple energy functions operating on different scales, the algorithm can focus on both low and high-resolution features in the image. By using multiple scales and energy functions, the algorithm can better capture the complex features of the input image and produce more realistic output.

5 Experimental Results

The code for the above all methods and training can be found here by authors of this paper at https://github.com/yilundu/improved$_{c}ontrastive_{d}ivergence$.
I have tried running some of this codes and will continue to do so in future as well for any possible novelty or implementations.

To show difference between simple contrastive divergence and proposed improved contrastive divergence, i found code at https://github.com/yilundu/improved$_{c}ontrastive_{d}ivergence$ and run the codes to get the actual loss difference when these 2 methods were implemented. I even tried and implemented training of restricted boltzman machines using contrastive divergence training of the energy-based algorithms. The code for same can be found at Unsupervised Deep Learning - Restricted Boltzman machine

5.1 The Experimental setup

I run both the training function for different instances of training data. The training data used was simple multivariate Guassian random variable n dimensional dataset. where the mean of the dataset was set to 0 and varience of the dataset was set to 1. Roughly 1000 training samples were used for the experiment. The batch size of 10 was used with each epoch of size 10. The learning rate for both the sample training functions used was 0.01.

5.2 Key observations

I observed that the loss at the initial epochs were quite same for improved contrastive to contrastive divergence, but as we slowly get into more and more epochs, I observed that the loss was reducing drastically for improved contrastive divergence compared to simple contrastive divergence.

6 Key benefits described in the paper

There are some key observations and empirical benefits of using different optimizations shown above.

6.1 Effect of data Augmentation

The authors empirically evaluated the effects of data augmentations and showed that by using downsampling, we can actually generate a diverse set of images instead of just converging to some set of points every time.

6.2 Mode convergence

The algorithms converged very faster, and improved stability or generalisability was observed among generations.

6.3 Out of distribution

Practically authors showed that instead of simple contrastrive divergence, they showed that by more efficiently exploring modes of energy distributions at training time, we are able to reduce the spurious modes of the energy function and thus improve out-of-distribution performance.

6.4 Compositionality

The authors showed that even we can use EBMs to combine with other Generative models for batter generative improvements. They trained different independent EBMs that learn different conditional distributions over concept factors. For example one EBM for nose generation, one for eyes generation etc. and then combining them showed batter generative abilities.

7 Future works

There are several future research directions in the area of energy-based models. Here are a few examples:

1. Scalability: Energy-based models are known to have scalability issues, particularly in high-dimensional spaces. Future work can focus on developing methods to improve the scalability of energy-based models, such as efficient training algorithms and architectures that can handle large datasets.
2. Integration with deep learning: While energy-based models have shown promising results in several applications, they have not yet been integrated fully with deep learning frameworks. Future work can focus on developing methods to combine energy-based models with deep learning architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs).
3. Robustness: Energy-based models can be sensitive to outliers and noise in the input data. Future work can focus on developing methods to improve the robustness of energy-based models, such as incorporating robust loss functions or using adversarial training.
4. Applications in reinforcement learning: Energy-based models have shown promise in the area of reinforcement learning, particularly in modeling the value function or the policy. Future work can focus on developing more advanced energy-based models that can handle more complex reinforcement learning tasks.
5. Interpretability: Energy-based models can be difficult to interpret due to their complex energy functions. Future work can focus on developing methods to improve the interpretability of energy-based models, such as developing visualization techniques or using attention mechanisms.

8 Conclusion

The improved contrastive divergence (CD) training of energy-based models (EBMs) is a promising approach to address the challenges of training EBMs on high-dimensional datasets. By using an adaptive learning rate and a dynamic batch size, the proposed method can achieve faster convergence and better sample quality than the traditional CD algorithm. The inclusion of the entropy loss

term further improves the diversity of generated samples. Overall, the proposed method has shown promising results on various image datasets and has the potential to be extended to other types of data. Future work can focus on improving the scalability and robustness of the proposed method and exploring its applications in reinforcement learning and interpretability.

References

[1] Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence

[2] Michael Arbel, Liang Zhou, and Arthur Gretton. Gen- eralized energy based models. arXiv preprint arXiv.org/abs/2003.05033, 2020

[3] Hinton, Training Pro ducts of Exp erts by Minimizing Contrastive Divergence, 2002

[4] Beomsu Kim, Jong Chul Ye, KAIST, Energy-Based Contrastive Learning of Visual Representations

[5] Guido Montufar, Restricted Boltzmann Machines: Introduction and Review

[6] Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle1, Liam Paull, Yuan Cao, Yoshua Bengio, Your GAN is Secretly an Energy-based Model and You Should Use Discriminator Driven Latent Sampling