Towards More Effective Energy-based Deep Learning

Prateek Malhotra
University of California, Los Angeles
prateekmalhotra@g.ucla.edu

***Abstract*— This paper proposes a new, novel architecture to learn smooth a energy function approximated using Convolutional Neural Networks. The algorithm uses Maximum Likelihood Estimation to learn the energy distribution of the entire training dataset using a modified version of contrastive divergence involving a separate generator network. The finite step MCMC sampling in our models are initialized using downsampled 1x1 pixels of the training images followed by upscaling of the sampled images. We reinitialize the MCMC algorithm at each step until we reach our final image size. Langevin dynamics for sampling from this generative model are driven by the reconstruction error of the training images and the sampled images; futhermore, the energy of the training examples is decreased in every iteration making their sampling more probable. This paper's experiments show that adding a generator network to the multiple grid approach leads to stable training, a smoother energy function, and empirically tested high-quality sampled images.**

I. INTRODUCTION

Generative models and unsupervised learning are tremendously useful for real-world problems where the dataset does not include labels for the training images. In such a setting, it becomes very important to learn features from the dataset which can then be used directly for specific tasks like classification and discriminative learning[13]. Generative modeling is also of paramount importance when the labels are scarce or the number of images are very small. The model can also be thought of as a tool for exponential tilting of a reference distribution which serves as a prior for the training[8]. In this paper, the distribution is taken to be Gaussian white-noise because it is the distribution with highest entropy given a value of variance [13].

The problem, however, is non-trivial because we are interested in directly sampling from the probability distribution of the distribution which means approximation of a very complex function using a very low number of images (relative to the dimensionality of data which for 64x64 RGB images means 12288 dimensions)[8]. Alternatively, we can also think of a generative network as being a piece-wise Gaussian where the filters in the bottom-up operations are the basis functions in the top-down representation. This fact, however, is only true if we choose Rectified Linear Unit (ReLU)[10] as our activation function in our Neural Network.

One major problem with energy based deep learning involves the highly multimodal estimation of the training dataset which this paper aims to conquer. The problem lies in the fact that sometimes the training data images are highly varied (the CIFAR 10 dataset which contains only 10 classes still remains a challenge for generative modeling)[8]. This introduces a severe bias in the sampling and the synthesized samples can be very different from our ideal distribution. To deal with this, we focus on using a modified version of Contrastive Divergence[5] where each image is passed through a generator network before it is able to initialize the finite step MCMC. Also, in the interest of a smoother function, we use a Multiple Grid approach[1] so that the model is less biased and less impacted by the individual variations between training examples.

Fig. 1: From left to right: the multiple grid approach initializes the first grid (1x1) from downsampled training images and the second grid (16 x 16) is initialized from the output of the previous one. This process continues until we reach our target image size[1]

After we obtain the synthesized images in each step, we update the Energy Function so that the energy of training examples passed through the Generator network stays lower than the locally sampled images[5]. The models at multiple grids[1] that synthesize these images are also trained using backpropagation to reduce the Mean Squared Error (MSE) between training images and sampled images at different levels.

The advantages of the proposed method are as follows:
(1) We introduce a Generator Network to reduce the subspace in the high-dimensional space from which images are sampled. Thus the learned energy function is smoother.
(2) Due to the generator network being used before the images passed to the multiple grid approach[1], we can say that the model is less biased by the high variations in features of multiple classes. This also encourages smoothness in the function as the grids move from coarser to finer.
(3) Thus, this paper's algorithm continuously builds on the previous grid and encourages MCMC mixing so that the samples drawn from the energy distribution are highly varied and photo-realistic.

II. RELATED WORK

This paper's work is related to the Contrastive Divergence (CD) technique introduced by Hinton et al. [5]. CD directly initializes a finite step MCMC from the training examples. In contrast, this paper approaches the problem by initialzing the MCMC process by passing the images through a generator network and thus, it is closer to Maximum Likelihood Estimation (MLE). Similar to CD [5], our aim is also to reduce Kullback-Liebler divergence between the true distribution and our approximation. However, in our method the true distribution is itself a smooth approximation of the original images.

Other than this energy based approach, we also have generator network which work by different methodologies like the Variational Autoencoder (VAE) [6] where the model is trained with an assisting model known as the inferential network. These dynamics can make the training unstable and usually there is no motivation in the learning procedure to map out the entire image space. Thus, energy based methods are more robust and powerful. A major focus of this paper is to learn energy based models while only working with simple Convolutional Neural Network architectures which work as function approximators[7].

Thus, the method is more generalizable than other generative methods which rely on hard approximations and constraints on the training data. Another major tool used in this paper is the Markov Chain Monte Carlo (MCMC) algorithm [2]. The method was originally introduced in statistical mechanics and it, along with simulated annealing, made its way into sampling methods and function approximations.

It may be said that our work directly builds on some recent work on generative modeling using energy based learning as introduced by [1] and [13]. Gao et al.'s [1] Multigrid approach is used here because we are primarily interested in a smooth distribution of energy which can lead to improved sampling as shown by [8].

Another class of models which compete against energy based deep learning are Generative Adversarial Networks (GANs) [12]. Generative Adversarial Networks, however, are highly unstable with respect to training and convergence. While our method introduces an added generator network, we can say that our training is much more stable as compared to GANs even without using constraints, modified loss functions and progressive training which are all intended to make GANs usable[4]. Hence, while the samples generated are not as photo-realistic, we build on the new paradigm of interpretable deep learning and Artificial Intelligence by using a method which is more transparent, intuitive and easier to diagnose. Thus, we learn deep energy based models to approximate probabilistic tendencies in the approximation of the training data by using an added generator network.

III. PROPOSED APPROACH

A. Notation

Following the convention laid down by [5], use $p(Y;\theta)$ or p_θ to denote the probability distribution of the energy based network and q_α to denote the probability distribution of the generator network which is trained cooperatively to approximate the training examples. α here denotes the parameters of the generator network.

B. The Energy Based Model

The energy based convolutional network can be defined as follows:

$$p_\theta(Y) = \frac{1}{Z(\theta)} exp[f_\theta(Y)]p_0(Y) \tag{1}$$

Here, p_0 denotes the reference distribution which, in this paper, is Gaussian white noise because it is the distribution obtained from entropy maximization. $p_0 \propto exp[-Y^2/2\sigma^2]$. f_θ is our energy based function that we approximate using a Convolutional Neural Network. The normalizing constant in (1) is denoted by $Z(\theta) = \int exp[f_\theta(Y)]p_0(Y)dy$ which is analytically intractable.
We can also rewrite the probability distribution as:

$$p_\theta(Y) = \frac{1}{Z(\theta)} exp(-E_\theta(Y)) \tag{2}$$

where E_θ is now the energy function learned using exponential tilting of the prior distribution.

C. Maximum Likelihood Learning

The Maximum Likehlihood method seeks to maximize the log-likelihood function denoted by:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log p_\theta(Y_i) \tag{3}$$

Using (3), the MLE minimizes the KL divergence: $KL(p_{data}||p_\theta)$ between our approximated distribution and the target distribution. Finding out the gradient of (3),

$$L^{'}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta f_\theta(Y_i) - E_\theta(\nabla_\theta(f_\theta(Y)) \tag{4}$$

Now, because the expectation term E_θ cannot be found out analytically, we apply MCMC methods: specifically, Langevin Dynamics [11], to approximate it using the training data. We can run multiple parallel chains of Langevin Dynamics to get generated examples from the original training data and these are denoted by $\hat{Y}$. Hence, the Monte Carlo approximation of (4) is written as:

$$L^{'}(\theta) = \nabla_\theta(\frac{1}{n}\sum_{i=1}^{n}(E_\theta(\hat{Y}_i)) - \frac{1}{n}\sum_{i=1}^{n}(E_\theta(Y_i))) \tag{5}$$

And only (5) is then used to update our Energy based model.

D. Contrastive Divergence [5]

The MCMC sampling technique introduced above will take a lot of time to ensure mixing because the training data is usually of a multimodal nature. Thus, we have to restrict ourselves to finite-step MCMC sampling. If we denote M_θ as the transition kernel of the finite step MCMC that samples from p_θ. Thus, $Mp(Y^{'}) = \int p(Y)M(Y,Y^{'})dY$

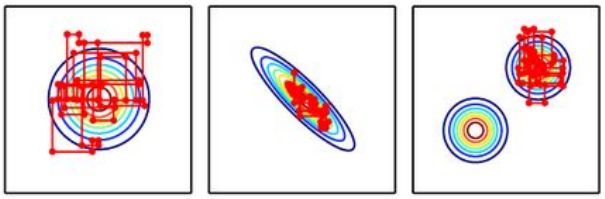


Fig. 2: Illustration of the multimodal distribution of the energy space and the difficulty in mixing of the MCMC method [3]

which denotes the marginal distribution.
Finally, we can say from [1] that the gradient from (5) approximates,

$$KL(p_{data}||p_\theta) - KL(M_\theta p_{data}||p_\theta) \tag{6}$$

p_{data} here is a very spiked distribution because it is only substantial in places where there exists a training example. This may lead to a bad approximation which leads to a bias in the CD estimate. The challenges still remain the same: exploring different modes of the distribution and Figure 2 illustrates this problem.

E. Dilated Convolutions

Dilated Convolutions [14] introduced by Fisher et al. has achieved state of the art on many semantic segmentation datasets. It is motivated by the fact that it supports exponentially expanding receptive fields without losing resolution or coverage. The following figure visually explains the concept of dilated convolutions. Each element in the output for 1-dilated kernel has a receptive field of 3x3 while for 2-dilated kernel and 4-dilated kernel the receptive field is 7x7 and 15x15 respectively (as described by Fisher et al). For our model, we apply 2-D dilated convolutions as follows:-

$$y[m,n] = \sum_{i=1}^{M}\sum_{j=1}^{N} x[m + r \cdot i, n + r \cdot j]w[i,j]$$

Where $y[m,n]$ is the output signal, $x[m,n]$ is the input signal and $w[i,j]$ is the weight associated with the convolution kernel of size $[M \times N]$. In the above formula, if r is set to 1, the dilated convolution becomes a regular convolution.

F. Generator Architecture

The architecture for the Generator network is very similar to the one suggested by [14] for the task of gray matter segmentation. This architecture was chosen specifically for its low number of parameters, small network size and distinct branches with different dilated kernel sizes. Their architecture also very recently achieved state-of-the-art in the gray matter segmentation challenge[**?**]. In this setup, the input image is first fed to a regular 3x3 convolution layer and

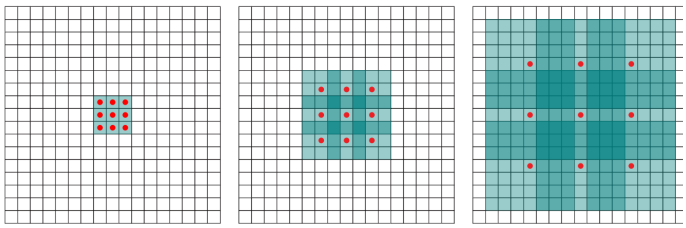


Fig. 3: From left to right: 1-dilated kernel, 2-dilated kernel and 4-dilated kernel. The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly[14]

then to a 3x3 convolution layer having a dilation rate of 2. This architecture achieved state of the art for crowd density estimation with a very low number of parameters [14].

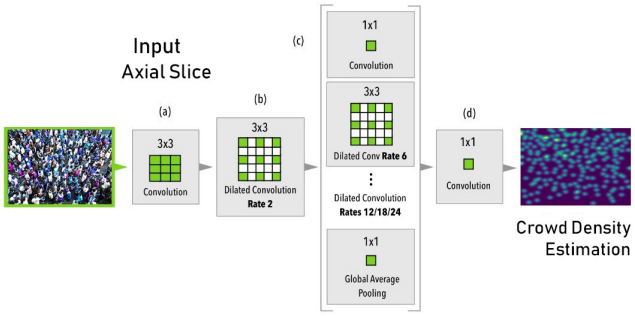


Fig. 4: In the above architecture, the input height and width dimensions are same as the output's height and width dimensions (H x W). Padding is used throughout to ensure that (h x w) of all intermediate feature maps is also (H, W) [14]. This network recently achieved state of the art in crowd density estimation.

After these steps, the resultant intermediate feature map is fed to six parallel branches containing convolutional layers of different dilation rates. The result is then concatenated and passed through a final 1x1 convolutional block. Since the output map is a probability density, we apply a ReLU [10] function to ensure that all pixels hold a value greater than or equal to 0.

G. The Multigrid Approach [1]

The Multigrid method proposed by Gao et al. works in the following way:
(1) Each training example Y_i is first downsampled from its original size (64 x 64) to a new size of (1 x 1).
(2) A finite step MCMC is initialized from this (1 x 1) downsampled image to get a new synthesized example. This synthesized example then serves as the starting initialization of the MCMC at the next grid (16 x 16).
(3) This step is repeated until the final image size is reached. After that, all grids are simultaneously updated to make sure that the synthesized examples are close to training examples in terms of Euclidean distance.

H. The Modified Multigrid Approach

In this paper, we propose to add a generator network to the Multigrid network presented above. The rationale is that a generator network cannot perfectly map all the training examples [8] and hence, produces a distribution where even distinct training examples are clustered closer in a smoother distribution.

This approach has a major advantage that the training examples are mapped into a much smaller subspace which is easier to approximate. Thus, in our method we are interested in reducing:

$$KL(P^{(s)}_{generator}||p^{(s)}_{\theta}) - KL(M^{(s)}_{\theta(s)}p^{(s)}_{\theta(s-1)}||p^{(s)}_{\theta(s)}) \quad (7)$$

I. Algorithm

(1) This papers Modified Multigrid Algorithm introduces an autoencoder network G (using the architecture presented above) to regenerate the training images.

(2) G is learned together with the Multigrid Network which tries to approximate it.

(3) The energy function of the image space gets progressively more complex (while retaining its smoothness) every epoch as G becomes more finely tuned.

Input:
(1) training examples $\{Y^{(s)}_i, s = 1, ..., S, i = 1, ..., n\}$
(2) number of langevin steps l (3) number of learning iterations T

Output:
(1) estimated parameters $(\theta^{(s)}, s = 1, ..., S)$,
(2) synthesized examples $\{\ \widetilde{Y}^{(s)}_i, s = 1, ..., S, i = 1, ..., n\ \}$

1: Let $t \leftarrow 0$, initialize $\theta^{(s)}, s = 1, ..., S$
2: **repeat**
3: For $i = 1, ..., n$, initialize $\widetilde{Y}^{(0)}_i = Y^{(0)}_i$
4: For $s = 1, ..., n$, initialize $\widetilde{Y}^{(s)}_i$ as the upscaled version of $\boldsymbol{G}(\widetilde{Y}^{(s-1)}_i)$ and run l steps of langevin dynamics to evolve $\widetilde{Y}^{(s)}_i$.
5: For $s = 1, ..., S$, update $\theta^{(s)}_{t+1} = \theta^{(s)}_t + \gamma_t L'(\theta^{(s)}_t)$, with step size $= \gamma_t$
6: $t \leftarrow t + 1$
7: **until:** $t = T$

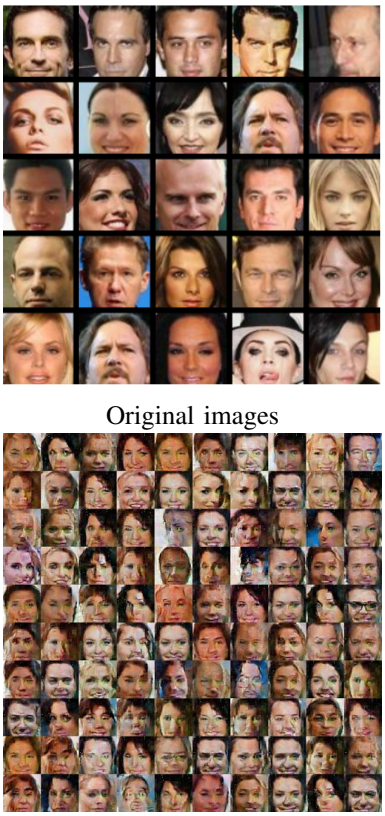
The only disadvantage is that an extra network is added to the system making training more complex as compared to the original multigrid method introduced by Gao et al [1].

IV. EXPERIMENTS

We test our approach on the one dataset which is widely used in the literature: The CelebA dataset [9] Figure 5 shows the original images and the corresponding generated images.

V. THE MODE COLLAPSE PROBLEM

As the energy function becomes more tuned beyond a certain threshold, the images being generated look progressively worse and worse. This is because the Langevin dynamics iterations slowly start to move away from regions that are near existing training examples. This remains an open problem with important implications such as solving overfitting in deep neural networks. The mode collapse results of this paper's network are uploaded to the author's github profile(github.com/prateekmalhotra). Empirically, this approach makes the model more robust and delays the mode collapse because of the smooth approximation introduced.



Original images

Generated Images

Fig. 5: Original and Generated Images from the CelebA dataset [9]

VI. CONCLUSION

In this paper, a novel method was introduced to obtain a smooth Energy function to map the probability distribution of the dataset in the image space for performing image-based unsupervised learning tasks. Our introduced generator network uses Dilated Convolutional layers to better approximate different spatial characteristics of the training dataset. We display our results on the CelebA dataset and empirically evaluate the method's mode collapse problem. This paper's philosophy of approximating another approximation encourages the MCMC method to mix better and generate diverse results. Furthermore, the features learned due to our energy based method can be directly applied to tasks like Image classification and other forms of discriminative learning.

REFERENCES

[1] Ruiqi Gao, Yang Lu, Junpei Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning generative convnets via multi-grid modeling and sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[2] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC, 1995.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[4] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

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

[6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] Yann LeCun, Sumit Chopra, and Raia Hadsell. A tutorial on energy-based learning. 2006.

[9] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15:2018, 2018.

[10] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[11] Richard W Pastor, Bernard R Brooks, and Attila Szabo. An analysis of the accuracy of langevin and molecular dynamics algorithms. *Molecular Physics*, 65(6):1409–1419, 1988.

[12] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[13] Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. Cooperative learning of energy-based model and latent variable model via mcmc teaching, 2018.

[14] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.