Direct Optimization of the Latent Representation for Fast Conditional Generation

David Acuna
University of Toronto
davidj@cs.toronto.edu

David Duvenaud
University of Toronto
duvenaud@cs.toronto.edu

Abstract

Gradient-based optimization of images has been proved to be a promising approach to generating state-of-the-art class-conditional images with crisp details and coherent overall structure. However, the optimization process is considerable more expensive than most class-conditional image generation techniques as the optimization needs to go through both the inference and the image generation networks. We present an alternative approach that eliminates the necessity to back-propagate through the composite architecture. In this way, we have turned the big-optimization problem into the optimization of a simple classifier from a low dimensional vector, which is additionally independent of the dimension of the image.

1 Introduction

Building good generative models of natural images has been a long standing challenge within the machine learning and computer vision communities. State-of-the-art research in the area has led to models capable of synthesized realistic images that capture the overall coherence of the pictures. These models are generally based on Generative Adversarial Networks (GAN) or Variational Autoencoders (VAE). While GAN and VAE's based models often work well at low resolutions (e.g. 32 x 32), they generally struggle to generate high-resolution images (e.g. 128 x 128 or higher).

A completely different approach, due to Nguyen et. al. [1], uses gradient-based optimization of images to do approximate *maximum a posteriori probability* (MAP) estimation, synthesizing large (227 x 227) class-conditional images. In this approach (figure 1a), an initial image is specified by a random vector $z \sim \mathcal{N}(0, I)$ which is passed through a pre-trained image generation network $g_\theta(.)$. The latent vector is then optimized according to the gradient of the chosen class probability with respect to the image , multiplied by the gradient of the image with respect to the latent vector, using backpropagation. The major limitations of this model are (1) the need for ad-hoc tweaks to the objective, and (2) the lack of diversity in the generated samples. While samples may vary slightly the whole image tends to have the same composition (e.g. a single plant with a green background).

In [2], Xu et. al. used the same architecture (figure 1a), but proposed to sample from $p(\text{image}|\text{class})$ using gradient-based MCMC methods. This approach produces realistic and content-diverse class-conditional images, and removes the need for *ad-hoc* tweaks to the objective when longer iterations are introduced. However, the slow convergence of MCMC methods particularly in high-dimensional data spaces [3] makes the optimization process more expensive.

In [4], Nguyen et. al. extended the method presented in [1] by introducing an additional prior on the latent code (figure 1b). This modification improves both sample quality and sample diversity, leading to a state-of-the-art generative model that produces large and diverse high quality images. They also demonstrated the ability of the model to generate images conditioning on caption when the inference network was replaced for a pretrained image captioning network.

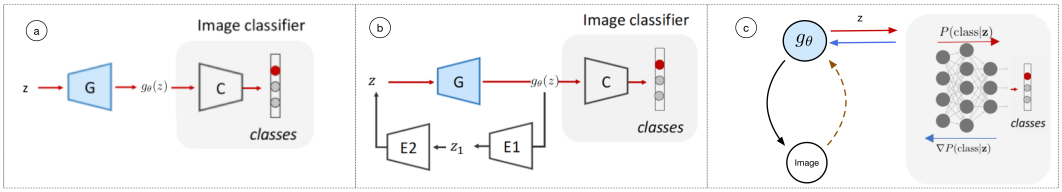

Figure 1: Different Gradient-Based Image Generation Architectures. (a) Model used in [1] and [2] where z is the latent code being optimized. G is the generator, and C estimates $p(\text{class}|\text{image})$. In this architecture, the authors respectively used MAP and gradient-based MCMC to synthesized large (227x227) class conditional images. (b) This architecture presented in [4] introduced an additional prior on the latent code leading to a state-of-the-art generative model that synthesized diverse class-conditional images. (c) Proposed architecture, we introduced a simple multi-layer perceptron trained to estimate the $p(\text{class}|z)$. The latent vector z is then optimized according to the gradient of the chosen class probability using back-propagation, eliminating the necessity to back-propagate through the composite architecture.

While the methods presented in [2] and [4] are able to produce realistic and content-diverse class-conditional images, their generation process is considerable more expensive than most class-conditional image generation techniques. This happens because in order to generate an image, the optimization needs to go throughout the composite architecture, meaning that we have to back-propagate through both the inference and the image generation network.

In this paper, we present an alternative architecture (figure 1c) that tries to mitigate the slow optimization problem. For this, we introduce a simple multi-layer perceptron (MLP) that outputs the $p(\text{class}|z)$ and then, instead of considering the optimization throughout the composite architecture, we used gradient based optimization to find the code z that maximizes the $p(\text{class}|z)$. In this way, we have turned the big-optimization problem into the optimization of a simple classifier from a low dimensional vector. It is also worth to mention that in our approach, the dimension of the optimization is independent of the dimension of the image.

2 Model Architecture

Like the previous presented architectures, our model uses two types of pre-trained neural networks. Firstly, we used an adversarially-trained image generation network $g_\theta(.)$ from [5]. This networks was trained to invert the $fc6$ layer of CaffeNet [6] and it takes in an input vector and outputs a synthetic image. Secondly, an off-the shelf image classification network (CaffeNet) is used to compute the $p(\text{class}|\text{image})$. Similar to [4], we also tested our model conditioned on caption. For that, we used a pre-trained version of LCRN [7], a two-layer LSTM network that generates captions conditioned on features extracted from the output layer of AlexNet [8]. Note that, our text-to-image architecture is pretty similar to the one described in figure 1c. The only difference is that we added the two-layer LSTM network at the end of the MLP.

As opposed to [1], [2] and [4], we introduced an MLP that outputs the $p(\text{class}|\text{z})$, where z is the latent variable being optimized. This latent vector z is initialized from a Gaussian distribution $z \sim \mathcal{N}(0, I)$ and then optimized using Stochastic Gradient Langevin Dynamics (SGLD) [9; 10]. At the end, the image is generated by feeding the optimized latent vector z into the generator.

Also similar to [2], we approximately sample from our class-conditional posterior:

$$\hat{z} \sim p(z|\text{class}) \propto p(\text{class}|z)p(z) \qquad \text{where} \qquad p(z) = \mathcal{N}(0, I) \tag{1}$$

However, in previous approaches the class conditional posterior is proportional to $p(\text{class}|g_\theta(z))$ and there is a necessity to back-propagate through both the inference and generator networks. In our approach, we only have to compute $\nabla \log p(\text{class}|z)$ reducing the complexity of the optimization and making it independently of the dimension of the image.

Note also that, inspired by SGLD, we define an approximate sampler by assuming small step size and doing away with the acceptance rejection step. The idea is that the stochasticity of SGD itself

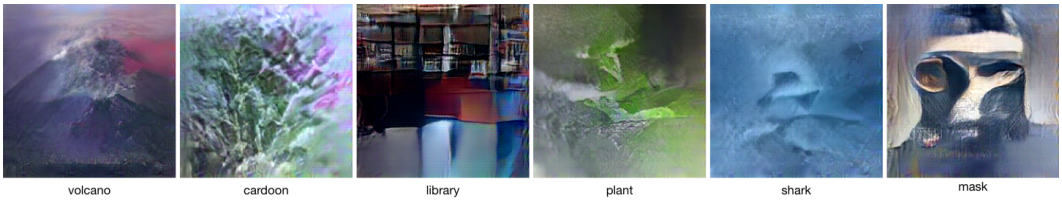

Figure 2: Class-Conditional Images synthetically generated using the presented approach

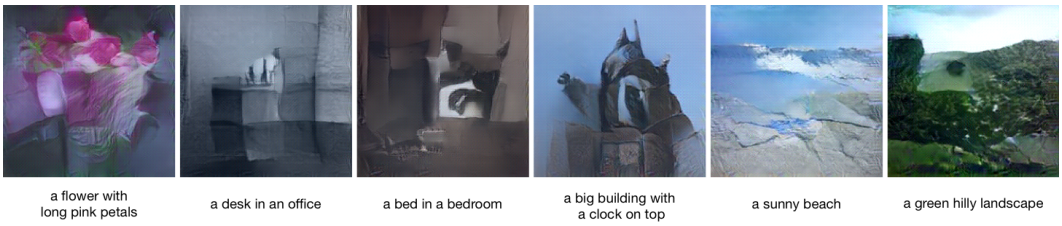

Figure 3: Text-to-Image generation

introduces an implicit noise, and while an acceptance ratio of 1 is only approached in the limit, it has been empirically observed that this approximation produces reasonable samples [4; 9; 10].

With this in mind, we defined a simple sampler with the following update rule:

$$z_{t+1} = z_t + \epsilon_1 \nabla \log p(\text{class}|z) + \mathcal{N}(0, \epsilon_2 I) \tag{2}$$

A major challenge of this approach is to find the latent code z corresponding to either a class-conditional or captioning image, so that we can train an MLP to output the probability of a specific condition given z. Note also that, the adversarially-trained image generation network is not invertible, and therefore, the architecture is not capable by itself to generate the corresponding vector z for a given image. However, the generator $g_\theta(.)$ was trained to invert the $fc6$ layer of AlexNet following an auto-encoder style [5] , and therefore, we can obtain a very similar approximation of z given the image, by feeding the image into the classification network and getting the output of the $fc6$ layer. In the case of the text-to-image architecture, we can follow a very similar approach as LCRN generates captions conditioned on features extracted from AlexNet.

3 Experiments and Results

3.1 Class-Conditional Image Generation

Our model was firstly evaluated conditioning on the ImageNet classes. For that, we used the architecture showed in figure 1c and the weights of the last three fully connected layers of AlexNet. These three fully connected layers actually constituted the MLP that outputs $p(\text{class}|z)$. The ϵ_1 and ϵ_2 values from equation 2 were empirically determined to be 1e-3 and 1e-8, respectively. As expected, the optimization was extremely fast compared to previous approaches. Figure 2 illustrates some of the synthetically generated images. Note from the generated images, that while our model is able to synthesized class-conditional images, the generated pictures are not as crispy as the ones presented in [4].

3.2 Text-to-Image generation

We also tested the performance of our model conditioning on caption. For that, we used an image-captioning recurrent network (LRCN) from [7] that was trained on the MS COCO dataset [11]. Since LCRN generates captions conditioned on features extracted from the output layer of AlexNet, we kept the MLP to be the last three fully connected layers of AlexNet, and we only added the LSTM to the architecture showed in 1c. We also kept the same optimization rule illustrated in equation 2, the only difference is that in the case of text-to-image generation, we firstly back-propagate through

the recurrent network and then through the MLP. Figure 3 illustrates some of the generated images conditioning on captions.

Similar to the class-conditional case, it is worth to mention that while our text-to-image architecture is able to synthesized images given captions, the generated representations lack the level of details of the ones obtained using the method presented in [4].

4 Limitations and Further Directions

Although our approach is able to generate high-resolution class-conditional images, and significantly improves the optimization speed, the generated images are not as crispy as the ones presented in [4]. Particularly, we noticed that the model struggles to generate high level of detail in small objects. This happens because the generator itself is not able to reconstruct that amount of detail in a real image, given an input code z from the features of AlexNet.We believe that further study using an invertible generator based on a Real NVP [12] or BiGan [13; 14] will mitigate the problem. In terms of text-to-image generation, it is important to highlight that similar to previous approaches, we often get mixed results, and while it works for some words, for others, it struggles to produce reasonable images.

5 Conclusions

We have presented a new method that is able to generate realistic and high-resolution class conditional images. Yet, it solves the slow optimization problem arising in previous approaches. For this, we introduce a simple multi-layer perceptron (MLP) that outputs the $p(\text{class}|z)$ and then, instead of considering the optimization throughout the composite architecture, we used gradient based optimization to find the code z that maximizes the probability of the chosen class. In this way, we have turned the big-optimization problem into the optimization of a simple classifier from a low dimensional vector.

References

[1] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," *arXiv*, pp. 1–29, 2016.

[2] B. Xu, D. Acuna, and D. Duvenaud, "Generating Class-conditional Images with Gradient-based Inference," *NIPS Workshop in Constructive Machine Learning*, Dec. 2016.

[3] R. M. Neal, "MCMC using Hamiltonian dynamics," *Handbook of Markov Chain Monte Carlo*, vol. 54, pp. 113–162, 2010.

[4] A. Nguyen, J. Yosinski, A. Dosovitskiy, and J. Clune, "Plug & Play Generative Networks : Conditional Iterative Generation of Images in Latent Space," no. 3.

[5] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," *CoRR*, vol. abs/1602.02644, 2016.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," *ACM International Conference on Multimedia*, pp. 675–678, 2014.

[7] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[9] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics.," in *ICML* (L. Getoor and T. Scheffer, eds.), pp. 681–688, Omnipress, 2011.

[10] Y. W. Teh, A. H. Thiery, and S. J. Vollmer, "Consistency and fluctuations for stochastic gradient langevin dynamics," *Journal of Machine Learning Research*, vol. 17, no. 7, pp. 1–33, 2016.

[11] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[12] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," *Arxiv*, pp. 1–29, 2016.

[13] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *CoRR*, vol. abs/1605.09782, 2016.

[14] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville, "Adversarially Learned Inference," *arXiv:1606.00704*, pp. 1–15, 2016.