A Survey of DCGAN based Unsupervised Decoding and Image Generation

Baomin Shao
Department of Computer Science and Technology, Shandong University of Technology
No. 266, Xincun Xi Road, Zibo, Shandong, China

Qiuling Li
Department of Computer Science and Technology, Shandong University of Technology
No. 266, Xincun Xi Road, Zibo, Shandong, China

Xue Jiang
Department of Computer Science and Technology, Shandong University of Technology
No. 266, Xincun Xi Road, Zibo, Shandong, China

ABSTRACT
At present, deep learning is a fast growing research area of machine learning, which can extract more effective features by using a cascade of nonlinear layer units, some of them use deep convolutional neural network to process digital images. DCGAN stands for Deep Convolutional Generative Adversarial Networks, which is an unsupervised learning method for neural networks. The model consists of two networks. A generator that generates an image from a noise vector, and a discriminator that discriminates between a generated image and a real image. The generator model takes random input values and transforms them into images through a deconvolutional neural network, whereas DCGAN generates an image from random parameters. In this paper, an unsupervised learning method is proposed which is based on DCGAN to generate the image of codes and use the discriminator to encode the image, in the training stage, except the minimax cost function for the GAN network, the distance between input code of generator and the output code of discriminator is minimized. On MNIST dataset, this method has achieved good experiment result, experiments show that this architecture can learn the code of images and further prove the method has the ability to understand a set of image without extra knowledge, and can reconstruct the image using the generated code.

General Terms
DCGAN, Image Encoding.

Keywords
DCGAN; image reconstruction; auto decoding.

1. INTRODUCTION
Images are 2D matrix data that carry visual information. They may take different forms in everyday life, such as video series, newspaper, book covers, movie posters, industry plans, company logos, or clothing design. It is a great challenge to process and analyze the massive amount of image data, each dataset often includes hundreds of images of same or different context, so intelligent image analysis is an important research direction to assist people in dealing big images. The best way to understand a specific dataset is to try to decode it and recreate it, comprehending images of the same kind usually follow a certain norm, following the whole process of deconstruction and reconstruction, which can be used to simulate the way intelligent agent understands the world. Before going to a pet shop, the portrait of dogs and cats can be imagined in head, with clear features such as size, color, shape, rather than the feature of car, plain, human, further, we can imagine several of dogs that have different features. This might seem obvious for human, because human have latent learning and training process, human vision ignores irrelevant details by using a carefully determined sequence of fixation points to ensure that only a tiny fraction of the optic array need to be processed, but not quite for computers, teaching a computer what's underlying the concept is not an easy task.

Generative models are one of the most popular approaches that are applied for this, as the model needs to be able to analyze and understand the essence of the training data before it can generate similar results itself[1]. Generative Adversarial Networks (GANs) [2] are proposed as dominating family of models to build good image representations and generate "realistic-looking" images. In GAN, the generator and discriminator are constructed with multilayer perceptrons. Built on top of that, DCGAN used Convolutional/Deconvolutional Neural Networks, which made the model more powerful. Deep Convolutional Generative Adversarial Networks (DCGANs)[3] are a class of neural networks that have gained popularity in recent years. They allow a network to learn to generate data with the same internal structure as other data. In this paper, a model of DCGAN for image rebuilding was built, unlike traditional DCGAN model, our model can maintain the input code in discriminator except the classification information, which means a more precised understanding the image during the reconstruction and feature extraction process.

The rest of this paper is organized as follows: Section II introduces several background techniques of GAN and DCGAN in our experiment. Section III describes the DCGAN autodecoder model and algorithm in detail. In Section IV, experiment results and discussion were given. Section VI makes some conclusion.

2. BACKGROUND
Recently Convolutional Neural Network (CNN) based approaches are becoming the dominant paradigm in almost every computer vision task. CNNs have shown outstanding results in various and diverse computer vision tasks such as stereo vision [4], image classification [5] or even difficult problems related with cross-spectral domains[6] outperforming conventional hand-made approaches. Based on a CNN, a fusion layer merges local information, dependent on small image patches, with global priors, computed using the entire image. The model is trained in an end-to-end fashion, so this architecture can process images of any resolution. However, finding a sufficient amount of labeled data is a major obstacle in machine learning classification tasks. Even though an increasing amount of data is becoming available on the Internet, majority of it is unlabeled. One way to benefit from the abundance of available data is to use unsupervised learning to learn reusable feature representations and then use

these intermediate representations on a variety of unsupervised learning tasks. Generative models are one of the most popular approaches that are applied for this, as the model needs to be able to analyze and understand the essence of the training data before it can generate similar results itself [2]. In the GAN framework[2], generative models are estimated via an adversarial process, in which simultaneously two models are trained: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a min-max two-player game. In the space of arbitrary functions G and D, a unique solution exists, with G recovering the training data distribution and D equal to 1/2 everywhere. In [7] some techniques to improve the efficiency of the generative adversarial networks have been proposed; one of them, referred to as the virtual batch normalization, allows to significantly improve the network optimization using the statistics of each set of training batches. Generative Adversarial Networks (GANs)[8, 9] have achieved impressive results in image generation[10, 11], and representation learning [12, 13]. Recent methods adopt the same idea for conditional mage generation applications, such as text2image [14], image inpainting [15], and future prediction [16], as well as to other domains like videos [17] and 3D data [18]. The key to GANs' success is the idea of an adversarial loss that forces the generated images to be, in principle, indistinguishable from real images.

for number of training iterations **do**
  for k steps **do**
    • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
    • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(\boldsymbol{x}^{(i)}\right)+\log\left(1-D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

  end for
    • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

Figure 1 The algorithm of GAN [2].

During GAN training, fix one model and upgrade parameters of the other model, G and D are iteratively trained competing against each other in a minimax game. Then, the distribution of sample data can be assessed by the generative model. GAN is very flexible and generic, and it can integrate different kinds of loss function.

As is discussed above, convolutional neural networks(CNN) perform really well on image classification tasks[19]. Utilize the GAN framework, CNN has been proven to be a suitablemodel for data generation. Deep convolutional generative adversarial networks(DCGAN) use convolutional neural networks as both the generator and the discriminator. By carefully choosing the architecture of CNN, DCGAN is able to train a good generator which learns a hierarchy of representations from dataset. Besides the CNN architecture, everything else is exactly the same as the GAN framework.

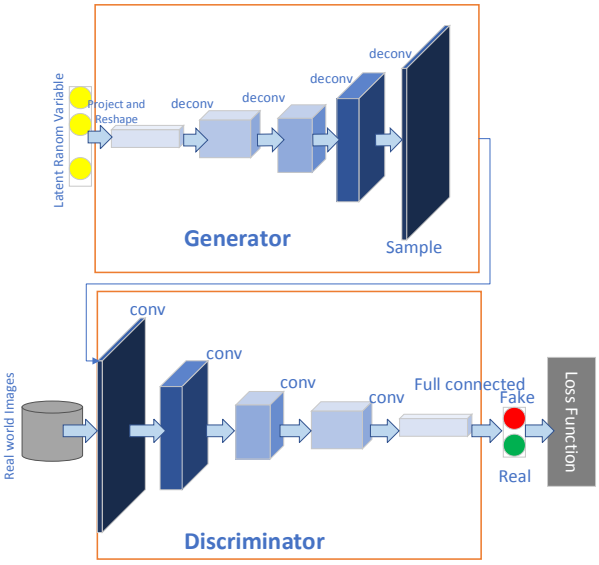

Figure 2 Architecture of DCGAN.

3. PROPOSED METHODS

3.1 Details of Network Model

This section presents the approach proposed for decoding and image generation. It is based on a traditional scheme of layers in a deep network. These models were often used to solve other types of problems such as feature extraction, similarity learning, etc. Based on these models, improvements in accuracy and performance have been obtained. The model we proposed receives as input the images to be able to generalize the learning process, the image dataset was normalized and an additive Gaussian Distribution noise was added to generate the necessary variability of the training set. A global loss function is used to minimize the overall classification error in the training set, which can improve the generalization capability of the model. The network is intended to learn to generate new samples from an unknown probability distribution. The DCGAN network has been trained using Stochastic AdamOptimazer since it prevents overfitting and leads to convergence faster. Furthermore, it is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters.

The architecture of the baseline model is conformed by convolutional, de-convolutional, relu, leak-relu, fully connected and activation function tanh and sigmoid for generator and discriminator networks respectively. Additionally, every layer of the model uses batch normalization for training. The following hyperparameters were used during the learning process: learning rate 0.0002 for the generator and the discriminator networks respectively; epsilon = 1e-08; exponential decay rate for the 1st moment momentum 0.5 for discriminator and 0.4 for the generator; weight initializer with a standard deviation of 0.00282; weight decay 1e-5; leak relu 0.2 and patch's size of 64×64. Figure 3 presents an illustration of the proposed DCGAN architecture.

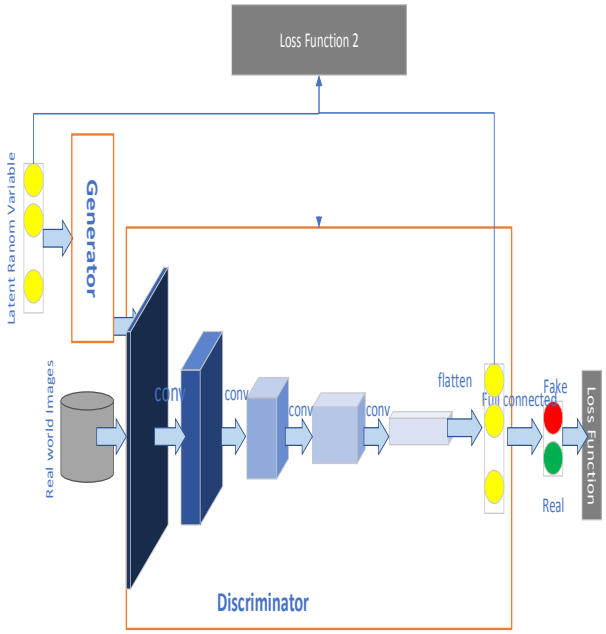

Figure 3 the architecture of proposed DCGAN model

3.2 Loss Function

The generator (G) and discriminator (D) are both feedforward deep neural networks that play a min-max game between one another. The generator takes as an input an image blurred with a Gaussian noise, and transforms it into the form of the data that are interested in. The discriminator takes as an input a set of data, either real image (z) or generated image (G(z)), and produces a probability of that data being real (P(z)). The discriminator is optimized in order to increase the likelihood of giving a high probability to the real data (the ground truth given image) and a low probability to the fake generated data. Normally, this procedure can be expressed as:

$$\min_G \max_D V(G,D) = \min_G \max_D (E_{x\sim p_{data}}[\log D_{\theta_d}(x)] + E_{z\sim p_z}[\log(1 - D_{\theta_d}(G_{\theta_g}(z)))]) \quad (1)$$

where D_{θ_d} is a multilayer perceptron that outputs a single scalar, p_z a prior on input noise variables z , G is a differentiable function represented by a multilayer perceptron with parameters θ_g . $D_{\theta_d}(x)$ represents the probability that x came from the data rather than from the generator's distribution.

The loss function was modified to better suit the purpose of the survey. The loss function can be expressed the addition of loss function and loss function 2 in Figure 3,

$$l_{total} = weight * l_{code} + (1 - weight) * l_{DCGAN} \quad (2)$$

In this paper we used an additional loss function to encourage the model to encode the instantiation parameters of the input data. Here l_{code} is the code loss between input distribution data and image encoding data, which can be calculated as the $l1$ norm or $l2$ norm or other measurement of their difference:

$$l_{code} = \left\| Code_{original} - Code_{decoded} \right\|_{1or2} \quad (3)$$

l_{DCGAN} is the adversarial loss and is the same as that in formula 1.

4. EXPERIMENTS AND RESULTS

The proposed model has been evaluated using MNIST dataset, after about 3000 iterations, the loss function tends to be steady, as is shown in the following figure.

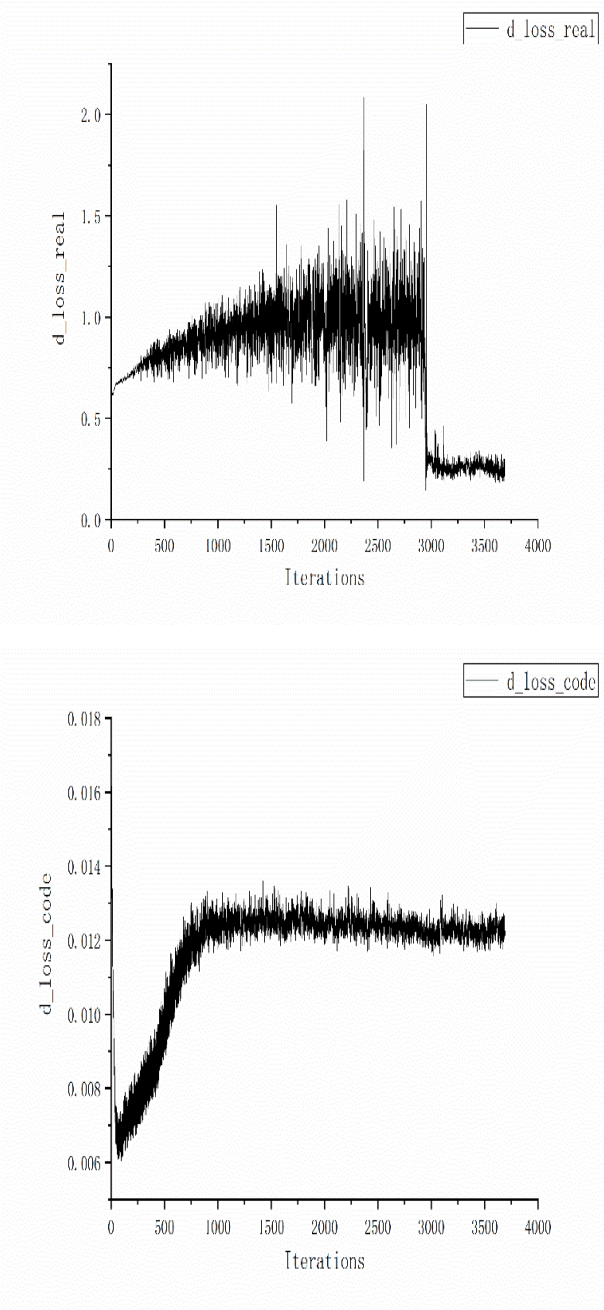

Figure 4 The loss value of evaluation real images and code difference

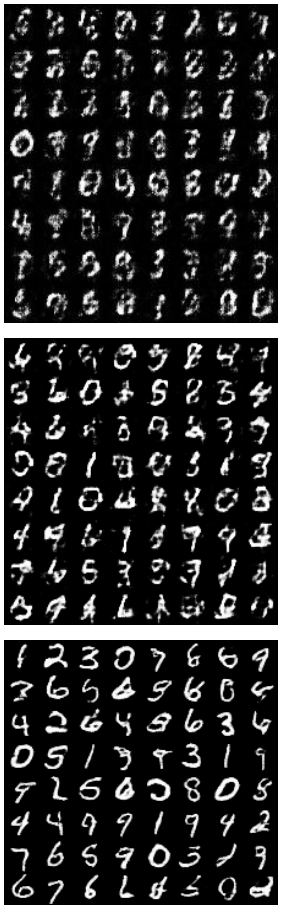

Figure 5 Generated number images after100, 200 and 2400 iterations of training.

During training, the sum of squared differences between the outputs of the discriminator code units and the input distribution units were minimized. We scale this code loss by 0.012 so that it is small enough and does not dominate the total loss during training, the change and comparison of loss are demonstrated in Figure 4. As illustrated in Figure 5. the generation of digits from the input distribution is robust while keeping the same generation latent space.

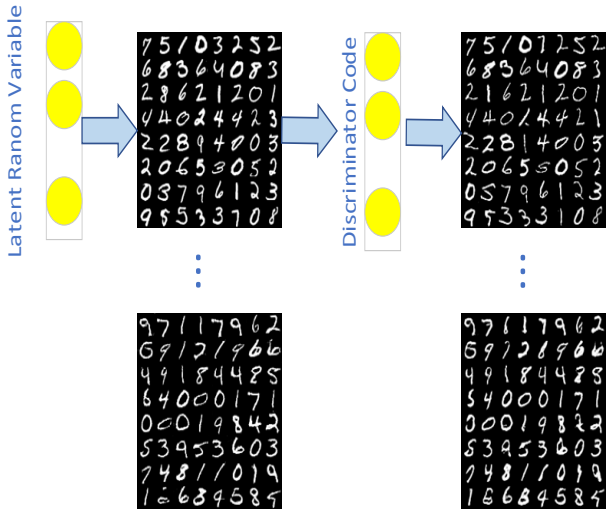

Figure 6 Digits generated from random variable and discriminator code

Figure 6 represents the input variable, generated digits, the discriminator code and the reconstruction target respectively. The two rightmost images show two reconstructions and it shows that model preserves the details and understanding of digits while smoothing the noise. explains how the model keep the same decoding of similar input data. The digit in the first row column 5 shows a failure example, it explains the model confuses a 3 and a 7 in this image, this is maybe due to the code of the two digits are so close in the latent space and the model can't tell the difference. Generally speaking, the digits from the original code and generated code can maintain not only the semantic information but also the digits morphology information.

In this paper, we built a modified DCGAN model which was able to generate images while keeping the code information. In this model the competition between the generator and the discriminator push the generator to produce images that look more appealing, the code loss function constrains the code position in latent space. The model, whose function was successfully tested, can generate images that are more appealing compared to conventional methods, the digits from the original code and generated code can maintain not only the semantic information but also the digits morphology information. Because DCGAN can learn from certain datasets, it can use trained features to produce images from inputs that lack certain information, the model can complete image details according to its understanding of the image dataset. For future work, one way to improve the results of reconstruction is to do a loop training, not just keep the loss function in 1 stage generating and classifying, but also extend it to multi stage procedure. Also, currently our work only discussed the proposed model on MNIST dataset, for real applications, we often have to deal with noisy low resolution images. Therefore, it might be great incentives to further investigate the model on degraded images.

5. REFERENCES

[1] OpenAI.GenerativeModelshttps://blog.openai.com/generative-models/, 2016.[Online; accessed 9-January-2018].

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

[3] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015.

[4] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *arXiv preprint arXiv:1510.05970*, 2015.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[6] C. A. Aguilera, F. J. Aguilera, A. D. Sappa, C. Aguilera, and R. Toledo. Learning cross-spectral similarity measures with deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2016.

[7] T. Salimans, I. Goodfellow,W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing*

Systems, pages 2226–2234, 2016.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014.

[9] J. Zhao, M. Mathieu, and Y. LeCun. Energybased generative adversarial network. arXiv preprintarXiv:1609.03126, 2016.

[10] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In NIPS, pages 1486–1494, 2015.

[11] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint rXiv:1511.06434, 2015.

[12] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. arXiv preprint arXiv:1606.03498, 2016.

[13] M. F. Mathieu, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun. Disentangling factors of variation in deep representation using adversarial training. InNIPS, pages 5040–5048, 2016.

[14] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396, 2016.

[15] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. CVPR, 2016.

[16] M. Mathieu, C. Couprie, and Y. LeCun. Deep multiscale video prediction beyond mean square error. ICLR, 2016.

[17] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In NIPS, pages 613–621, 2016.

[18] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In NIPS, pages 82–90, 2016.

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