Theoretical Analysis of Convergence and Associated Issues in Generative Adversarial Network (GAN) Using Evolutionary Algorithm

Soham Mandal, *Graduate Student Member, IEEE*
Under The Host Supervisions of
Prof. Dipti Srinivasan, *Senior Member, IEEE* and Prof. Kay Chen Tan, *Fellow, IEEE*
Under The Home Supervision of
Prof. Sushmita Mitra, *Fellow, IEEE*

***Abstract*—Being a black box model, one of the most prominent problem of deep learning research is the uncertainty of achieving convergence towards the global optimum solution. In this project report, first this problem has been discussed. Then considering the case of generative adversarial network (GAN), the theoretical analysis of convergence has been elaborated with the aid of evolutionary algorithms. Also a brief analysis on the evolutionary algorithm as reinforcement learning policies has been explored. The goal of this project is to explore the pathway for the development of a hybrid machine learning paradigm.**

***Index Terms*—Deep Neural Network, Gradient Descent, Generative adversarial network, Generator, Discriminator, Nash equilibrium, Reinforcement learning, Evolutionary algorithm, Differential Evolution, Evolutionary strategy.**

I. INTRODUCTION

DEEP learning [1] research has seen unprecedented growth in the recent years in a plethora of fields. But it is widely considered to be a black box model. There are multiple justified reasons behind such an inference about deep learning. The main factor is the fact that the convergence in the very deep neural networks are really difficult to achieve [2]. In general, gradient based techniques [2] are used to train the deep neural network. There are hardly any strong theoretical basis for justifying the practical tricks, which are used in training and constructing deep nets. No theoretical guarantee of reaching desired global optimum solution could be claimed in such training, unless some specific conditions (example: Robbins-Monro, etc.) are enforced, which heuristically motivate the network to converge. There are always high probability of getting stuck in a local optimum solution.

As there are numerous varieties of deep neural network, this study is specifically focused on the convergence issues of the generative adversarial network (GAN)[9]. It is a very recently proposed promising example of the generative model family. In general, a typical generative model is capable of learning the distribution of the training set to estimate the distribution explicitly and/or to generate samples from the learned distribution. Most of the variants of generative adversarial networks concentrate on the task of generating samples.

In this project report, at first the possibilities of utilizing different evolutionary algorithms for training a neural network have been explored. The specific advantages and disadvantages of training a neural network with this family of algorithms, which are observed in the work, have been noted. After that the theory of GAN has been described very briefly with certain remarks about its architecture. Then the theoretical analysis of the convergence in GAN through training using evolutionary algorithm has been described. Next the idea of treating reinforcement learning policy [2] as evolutionary computation techniques is explored from a theoretical viewpoint. In the final section, a conclusion is drawn with directions towards future extension of this project.

II. NOTES ON NEURAL NETWORK TRAINING USING EVOLUTIONARY ALGORITHM

In the neural network, the resultant output of the network can be considered as the function of the inputs x and the connection weights W. Here the idea is to minimize the error function, which itself is a function of the input and the output of the network. Thus from the optimization viewpoint, the error function is the objective function. Therefore, this optimization goal is achieved by tuning the weights of the network, as the initial inputs to the network will be fixed.

We have followed mainly few different ways [3][4][5] [6][7] of training the network utilizing evolutionary algorithms. For the sake of simplicity, we have restricted the studies to just simple multi layer perceptron (MLP) for exploring different attributes of training by evolutionary algorithms.

Quantitatively in all of the approaches, there are no significant improvement over the standard gradient descent. Not only this, but also the neural network seems to suffer from learning stagnation from training in those schemes. Still from the observation, it can be concluded that evolutionary algorithm could be used to train neural networks when the error

S. Mandal is with the Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, India. e-mail: smandal1215@gmail.com

D. Srinivasan is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. e-mail: dipti@nus.edu.sg

K. C. Tan is with the Department of Computer Science, City University of Hong Kong, Hong Kong. e-mail: kaytan@cityu.edu.hk.

S. Mitra is with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata. e-mail: sushmita@isical.ac.in

surface tends to be very rough, leading to infrequent gradient information.

In gradient descent, it is not possible to use differentiable function as the error function while training. Sometimes for certain error surfaces, it might be easier to use non-differentiable function for training the neural network. In the scheme of training using differential evolution there are no such strict restrictions over the error function. Here non-differentiable functions may be used.

Also as evolutionary algorithm does not get affected over the rate of descent with respect to the corresponding gradient, it remains unaffected from the occurrence of saddle points, which could be a significant detrimental factor for gradient based learning [8].

A. Notes on Generative Adversarial Network (GAN) Architecture

The unprecedented generative success of generative adversarial network [9] among all the state-of-the-art generative models [10] [11] [12] [13] can be attributed to its game theoretic base. It constructs the generative model by constituting a two player minimax game [14]. A generator network, being one of the players, seeks to generate synthetic data samples from the input noisy samples, drawn from a predefined distribution. Meanwhile the discriminator network, the other player, will go on to distinguish between the generated synthetic results and the samples, drawn from a real dataset. Based on the feedback from the discriminator network, the generator network will be continuously trained to fool the discriminator network to make it believe the synthetic samples being no different from the real samples. Generally massive deep learning architectures [1] are used as discriminator and generator networks [15] [16].

Mathematically, both players are represented by two differentiable functions. The discriminator can be represented as a function D which takes x as input and its parameters can be described as $\theta^{(D)}$. Analogously the generator can be portrayed as a function G, which takes z as input and its parameters can be defined as $\theta^{(G)}$. The cost of the discriminator can be defined as,

$$\begin{aligned} J^{(D)}(\theta^{(D)}, \theta^{(G)}) = &-\frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)} \log D(x) \\ &-\frac{1}{2}\mathbb{E}_{z\sim p_z(z)} \log(1 - D(G(z))). \end{aligned} \tag{1}$$

Here $p_{data}(x)$ and $p_z(z)$ are the training data distribution and generator input noisy distribution respectively. A simplified version of the proposed game is zero sum game. So the cost of the generator can be expressed as following,

$$J^{(G)}(\theta^{(D)}, \theta^{(G)}) = -J^{(D)}(\theta^{(D)}, \theta^{(G)}). \tag{2}$$

It can be easily perceived that the discriminator cost expression is nothing but the standard cross-entropy cost which is minimized when training a standard binary classifier with a sigmoid output. For better generalization, it is trained on two minibatches of data from the real dataset (the corresponding labels are 1) and from the synthetic generated samples (where the corresponding labels are 0). Thus the discriminator will always try to minimize its cost, while having the control of only $\theta^{(D)}$. In parallel the generator will look forward to minimize its own cost by controlling only $\theta^{(G)}$. Therefore each player's cost depends on the other player's parameters, though each player cannot manipulate the other player's parameters to gain advantage. Goodfellow et al. [9] claimed that instead of expressing this ensemble as an optimization problem between the agents, it would be more straightforward to portray this as a game between these two players, which would have a solution as a Nash equilibrium [14]. So the game would terminate at this saddle point, which would be a local minimum of $J^{(D)}$ with respect to $\theta^{(D)}$ and a local maximum of $J^{(G)}$ with respect to $\theta^{(G)}$. Therefore this complete game theoretic framework can be conceived as the following value function for the game,

$$\underset{\theta^{(G)}}{min}\ \underset{\theta^{(D)}}{max}\ V(\theta^{(D)}, \theta^{(G)}) = -J^{(D)}(\theta^{(D)}, \theta^{(G)}). \tag{3}$$

Goodfellow et al. [9] have shown that the learning of both those networks closely resembles minimizing the Jensen-Shannon divergence between the training data and the model distribution.

B. Theoretical Analysis of Convergence in GAN Through Evolutionary Algorithm

To analyze the convergence of generative adversarial network theoretically, our strategy would be to consider value functions of the discriminator and the generator separately. Now the goal is to show p_g converging to p_{data}, where p_g is the generator's distribution over x. Here instead of considering training by the evolutionary algorithm in general, we will consider a specific algorithm variant from evolutionary algorithm family. It is considered for this study that for the standard training of the GAN, the periodic updates to the weights of both the generator and the discriminator networks are done by differential evolution algorithms. As there are multiple prominent works [20][21], which have already been done exploring the convergence properties in differential evolution, here the objective of selecting differential evolution is to exploit those studies.

As both the networks continue to update with respect to each other's outputs with time, so the whole network would be in dynamic state continuously. Both the value functions are complex multimodal functions. So in case of gradient descent, there are high probability of getting stuck to local minimas [2], which might lead to non convergence of the whole network, even if the weight training is carried out for infinite amount of time.

In case of using differential evolution based training, if theoretically the training is carried out for infinite amount of time, then Ghosh et al. [20] have shown that the probability density functions (PDF) of the trial solutions (optimum set of weights benefiting the network's goal) would focus over the global optimum of the objective function. Here they have assumed that the function would take the shape of a Dirac Delta distribution [23]. They have proved the same by defining a Lyapunov functional [22] based over the PDF, which monotonically decrease.

Now from eq. 3, it can be seen that this is working over parametric space strictly. So to prove the convergence of the network by the scheme described by Ghosh et al., the value function needs to be converted to explicitly work in the space of the probability density functions of the generator (or discriminator).

The value function of the network in eq. 3 is written over the cost of discriminator. Therefore to achieve the above aim, first the optimal discriminator function D is required by keeping the generator function G fixed. Now as explained in the section II.A, to get the optimal discriminator, $V(\theta^{(D)}, \theta^{(G)})$ needs to get maximized for a fixed generator G. So the value function will be

$$V(\theta^{(D)}, \theta^{(G)}) = \int_x p_{data}(x) \log(D(x))dx + \int_z p_z(z) \log(1 - D(G(z)))dz. \tag{4}$$

Please note that here the $-\frac{1}{2}$ term has been avoided for the sake of convenience in calculation. It can easily be shown that this function becomes maximum at $D_G(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$. Here the generator G defines the probability distribution p_g, when $z \sim p_z$.

So from eq. 3, it can be said that value function of the minimax game between the generator and the discriminator network could also be written as

$$\underset{\theta^{(G)}}{min}\ \underset{\theta^{(D)}}{max}\ V(\theta^{(D)}, \theta^{(G)}) = \mathbb{E}_{x \sim p_{data}(x)}\left[\log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}\right] + \mathbb{E}_{x \sim p_g(x)}\left[\log \frac{p_g(x)}{p_{data}(x) + p_g(x)}\right] \tag{5}$$

In this manner by converting the value function of the network into the space of the probability density functions, it can be shown to achieve convergence theoretically by the scheme described above.

Even though, it sounds lucrative to train GAN with DE based approaches, but it might not be practically feasible. Because the inherent consideration for the theoretical guarantee of convergence is the continuous training for infinite amount of time. So going with our explored works, the resultant remark would be to train the GAN with variants of gradient based training only. But from our analysis, it can be inferred that the convergence can be achievable in every instances of GAN through the training by differential evolution algorithms.

C. Theoretical Justification of Heuristical Tricks Used in GAN Training

Also the heuristic tricks described by Sallisman et al. [18] mainly stresses on subdividing/modifying the goal of the optimization task. The most important and highly used two tricks are feature matching and historical averaging. Feature matching is changing the goal of generator to generate data based on only some features isolated by the discriminator. In historical averaging, the cost functions of both the networks at a particular iteration get modified depending on the values of the parameters occurred at every previous iterations of the training.

Now in a gradient based learning system, there are no strong theoretical justification towards adapting these heuristical modifications. But in a evolutionary algorithm based training scheme, it could be pretty straight forward to claim that feature matching has been done to favor the strong traits of the current generations to imbibe in the future generations of the solution set. In a similar fashion, historical averaging could be considered as a modified mutation strategy to achieve higher fitness.

D. Reinforcement Learning Simulation

In the next phase, evolutionary computation has been explored as the substitute to Markov decision process (MDP) based reinforcement learning (RL) policies [19]. Evolutionary strategy (ES) has been used as the alternative to reinforcement learning policies in this case. No strong theoretical motivation could be found from this study. But some prominent heuristical advantages have been observed such as, ES is invariant to action frequency and delayed rewards in similar tasks. Meanwhile ES does not need value function approximation, unlike standard reinforcement learning policies. It was evident that the high parallelizing capability of evolutionary strategy would be the main reason to support this scheme of simulation RL policies. Still if a MDP contains GAN as an actor critic method [17], it would be possible to assess its convergence theoretically from the above studies.

III. Conclusion

Though evolutionary algorithms do not enjoy any quantitative advantages over gradient based methods for training neural networks, but we have shown that they have been able to give better theoretical justification towards guaranteeing its convergence, which is highly vital to understand the inner workings of deep neural networks. We have also explored some other specific pitfalls of evolutionary algorithm based training. Further we have analysed the connection between GAN and reinforcement learning, with evolutionary algorithm being a substitute of reinforcement learning policies.

In future, we are looking forward to extend this work in some promising ways. First, we will try to extend our theoretical analysis in a more rigourous mathematical form for other variants of deep neural networks to achieve generalized solutions. It can be realized from a practical viewpoint that both gradient based learning and evolutionary algorithm based training are not suitable for every problems. Thus we will try to design a hybrid learning algorithm, which would be an amalgamation of both evolutionary algorithms and gradient based algorithms. This would utilize favorable traits from both the areas for better search towards convergence. Most importantly, for the sake of simplicity, we have restricted our studies of convergence of deep neural networks over the theoretical side only. So we will try to generate practical simulations for these cases, which would be congruent to our developed theoretical analysis of convergence.

ACKNOWLEDGEMENT

This research has been supported by IEEE Computational Intelligence Society Graduate Student Research Grant 2017.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, ”Deep learning”, Nature, vol. 521, no. 7553, pp.436-444, 2015.

[2] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning, MIT press, 2016.

[3] J. Ilonen, J. K. Kamarainen, and J. Lampinen, ”Differential evolution training algorithm for feed-forward neural networks”, Neural Processing Letters, vol. 17, no. 1, pp.93-105, 2003.

[4] A.P. Piotrowski, ”Differential evolution algorithms applied to neural network training suffer from stagnation”, Applied Soft Computing, vol. 21, pp. 382-406, 2014.

[5] G. Morse, and K.O. Stanley, ”Simple Evolutionary Optimization Can Rival Stochastic Gradient Descent in Neural Networks”, In Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, pp. 477-484, 2016.

[6] N. Leema, H.K. Nehemiah, and A. Kannan, ”Neural network classifier optimization using differential evolution with global information and back propagation algorithm for clinical datasets”, Applied Soft Computing, vol. 49, pp.834-844, 2016.

[7] H.A. Abbass, ”An evolutionary artificial neural networks approach for breast cancer diagnosis”, Artificial intelligence in Medicine, vol. 25, no. 3, pp.265-281, 2002.

[8] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, ”Identifying and attacking the saddle point problem in high-dimensional non-convex optimization”, arXiv preprint arXiv: 1406.2572, 2014.

[9] 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 27, pp. 2672-2680, 2014.

[10] R. Salakhutdinov, and G. E. Hinton, ”Deep Boltzmann machines”, In Twelfth International Conference on. Artificial Intelligence and Statistics, pp. 448455, 2009.

[11] Y. Bengio, E. Thibodeau-Laufer, and J. Yosinski, ”Deep generative stochastic networks trainable by backprop”, In Proceedings of the 30th International Conference on Machine Learning, 2014.

[12] D. P. Kingma and M. Welling, ”Auto-encoding variational bayes”, In Proceedings of the International Conference on Learning Representations, 2014.

[13] Z. Tu, ”Learning Generative Models via Discriminative Approaches,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-8, 2007.

[14] A.K. Jagannatham, and V. Kumar, ”Introduction to Game Theory. In Decision Sciences: Theory and Practice”, CRC Press, Taylor & Francis Group, pp. 79-144, 2016.

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

[16] E.L. Denton, S. Chintala, and R. Fergus, ”Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks”, In Advances in Neural Information Processing Systems 28, pp. 1486-1494, 2015.

[17] D. Pfau, and O. Vinyals, ”Connecting generative adversarial networks and actor-critic methods”, arXiv preprint arXiv:1610.01945, 2016.

[18] 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 29, pp. 2226-2234, 2016.

[19] T. Salimans, J. Ho, X. Chen, and I. Sutskever, ”Evolution Strategies as a Scalable Alternative to Reinforcement Learning”, arXiv preprint arXiv:1703.03864, 2017.

[20] S. Ghosh, S. Das, A. V. Vasilakos, and K. Suresh, ”On Convergence of Differential Evolution Over a Class of Continuous Functions With Unique Global Optimum”, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 42, no. 1, pp. 107-124, 2012.

[21] R. Knobloch, J. Mlynek and R. Srb, ”The classic differential evolution algorithm and its convergence properties”, Applications of Mathematics, vol. 62, no. 2, pp.197-208, 2017.

[22] W. Hahn, Theory and Application of Lyapunovs Direct Method. Englewood Cliffs, NJ: Prentice-Hall, 1963.

[23] W. Rudin, Functional Analysis, 2nd ed. New York: McGraw-Hill, 1991.