REG-GAN: SEMI-SUPERVISED LEARNING BASED ON GENERATIVE ADVERSARIAL NETWORKS FOR REGRESSION

Mehdi Rezagholizadeh[*], *Md Akmal Haidar*[†]

Noah's Ark Lab, Montreal Research Center, Huawei Technologies

ABSTRACT

This research concerns introducing a method to solve the semi-supervised learning problem with generative adversarial networks (GANs) for regression. In contrast to classification, where only a limited number of distinct classes is given, the regression task is defined as predicting continuous labels for a given dataset. This method will be of particular interest for the applications in which a small number of labeled samples is available, and the labels are continuous such as predicting steering angles from the front camera image in the end-to-end task of autonomous driving. Semi-supervised learning is of vital importance for the applications where a small number of labeled samples is available, or labeling samples is difficult or expensive to collect. A case in point is autonomous driving in which obtaining sufficient labeled samples covering all driving conditions is costly. In this context, we can take advantage of semi-supervised learning techniques with groundbreaking generative models, such as generative adversarial networks. However, currently almost all proposed GAN-based semi-supervised techniques in the literature are focused on solving the classification problem. Hence, developing a GAN-based semi-supervised method for the regression task is still an open problem. In this work, two different architectures will be proposed to address this problem. In summary, our introduced method is able to predict continuous labels for a training dataset which has only a limited number of labeled samples. Moreover, the application of this technique for solving the end-to-end task in autonomous driving will be presented. We performed several experiments to evaluate our proposed method, and the results are very promising compared with the state-of-the-art Improved-GAN technique [1].

Index Terms— Generative Adversarial Networks, Semi-Supervised Learning, Regression

1. INTRODUCTION

Autonomous driving (AD) has gained attention from researchers and industry in the recent years. To make the driving task autonomous, the AD system should replace human beings, which implies that the system should be able to recognize its surrounding environment and act accordingly. Machine learning can facilitate this task for the AD system by contributing to imitating driver behavior [2], vehicle detection [3], lane detection [3], and end to end learning [4]. Training AD systems needs a large number of training samples; on the other hand, collecting enough training samples and labeling them can be time consuming, difficult and costly. One solution to this problem can be using generative models for generating samples from a small unlabeled training set [5].

Generative models aim at estimating the probability distribution of the training data and being able to generate samples which belong to the same data distribution manifold [6]. Different methods for deep generative networks are proposed in the literature such as Deep Belief Networks (DBNs) [7], Restricted Boltzman Machines (RBMs) [8], Variational Auto-encoders (VAEs) [9], and GANs [10]; among them, GANs are the most recent and successful in generating realistic and good quality images [11]. However, pure unsupervised generative models are not able to label their generated samples. To resolve this issue, generative models can be used in a semi-supervised fashion. In the literature, there are some semi-supervised techniques with GANs such as Improved-GAN [1], Cat-GAN [12], SGAN [13], and Triple-GAN [14]; however, they all focus on solving the classification problem. On the other hand, the goal of the end-to-end task in AD is to predict the steering angle, which is a continuous variable, based on the given input image from the front camera.

Applying semi-supervised classification techniques to regression comes with the price of converting continuous labels of the dataset to a number of classes. This conversion will add the quantization error to our training. Also determining the number of classes for each application is non-trivial. To the best of our knowledge, a semi-supervised regression technique with generative adversarial network has yet to be developed which is the main focus of this research. The main contributions of this paper are summarized as follows:

1. To the best of our knowledge, it is the first semi-supervised algorithm with generative adversarial networks which can address the regression problem.

2. Applying semi-supervised classification techniques to

[*]The authors contributed equally to this work. mehdi.rezagholizadeh@huawei.com

[†]The authors contributed equally to this work. md.akmal.haidar@huawei.com

regression comes with the price of converting continuous labels to a limited number of classes. Our proposed approach avoids this quantization error and reduces the hyper-parameter that comes with it.

3. Our approach generates high quality images, smaller label prediction error and more stable training compared with the state-of-the-art Improved-GAN technique [1].

The remainder of the paper is organized as follows. In the next section, related work in the literature will be reviewed. Then, some preliminary background on generative adversarial networks will be described. The proposed method will be presented in section 4. In section 5, the results of the experiments will be shown. Finally, section 6 will conclude the paper.

2. RELATED WORK

In this section, we will review some relevant work to the idea of this paper in generative adversarial networks and semi-supervised learning. Semi-supervised learning techniques based on deep generative networks target improving the supervised task by learning from both labeled and unlabeled samples [15]. Semi-supervised learning is beneficial when labeled samples are not easy to obtain, and we have a small set of them but more unlabeled data. Using deep generative models (and generative adversarial networks recently) in semi-supervised learning has introduced remarkable improvements to the field [15, 1, 13, 16, 12, 17, 18, 19, 14]. For example, the Improved-GAN technique shows competitive test errors and high quality generated samples over MNIST, CIFAR-10, SVHN and ImageNet datasets. In contrast to Improved-GAN, Triple-GAN [14] employs two separate networks for classification and discrimination, thus having more parameters to learn, and increasing the training difficulty. However, almost all of these techniques target the classification problem and if we want to apply them to a regression problem, we need to quantize the continuous labels to a limited number of classes.

3. GENERATIVE ADVERSARIAL NETWORKS

Generative adversarial networks include two separate deep networks: the generator and the discriminator. The generator takes in a random variable z with distribution $p_z(z)$ and maps it to the data distribution $P_{data}(x)$. The discriminator is expected to discern real samples from generated samples by outputting of 1 or 0 respectively. In the GAN training process, the generator and discriminator are used to generate samples and classify them respectively by improving the performance of each other in an adversarial manner. In this regard, an adversarial loss function is employed in training the generator and discriminator [10]:

$$\begin{aligned}&\min_G \max_D E_{x\sim P_{data}(x)}[\log D(x)] \\ &+ E_{z\sim P_z(z)}[\log(1-D(G(z)))].\end{aligned} \tag{1}$$

The original GAN technique is not able to predict the label of the generated samples. The beauty of the Improved-GAN method [1] is to combine the task of classification and discrimination into the discriminator network, i.e. using one network for performing the two tasks. The Improved-GAN modifies the architecture of the discriminator to have N+1 outputs, where N represents the number of classes in the training dataset. The first N outputs should predict the probability of an input to belong to each class, $p(y|x, y < N+1)$; and the last output represents the probability of the sample to be fake $p(y = N+1|x)$. Moreover, the Improved-GAN uses the feature matching technique to address the instability issue of the generator. In contrast to traditional GAN techniques which have the generator try to maximize the output of the discriminator for its samples, feature matching tries to maximize the matching between the statistics of the generated and real samples inside the discriminator:

$$L_{\text{feature\_matching}} = ||E_{x\sim p_{\text{data}}} f(x) - E_{z\sim p_z(z)} f(G(z))|| \tag{2}$$

where $f(x)$ represents the output of an activation function of an intermediate layer of the discriminator.

4. METHODOLOGY

The core idea of our work is inspired by the Improved-GAN technique [1], and we try to extend Improved-GAN to be able to cover regression as well. The proposed method is comprised of a generator, which is responsible for generating realistic samples visually similar to the samples in the training dataset, and a discriminator, which is responsible for both validating the generated samples and predicting their continuous labels. The generator is trained by employing the feature matching loss technique (see Eq. 2).

We propose two architectures for the discriminator in our Reg-GAN method (see Figs. 1 and 2). In the first approach, the discriminator is built with two outputs: one is responsible for predicting the label, and the other for predicting the probability that the generated sample is real/fake. If we assume that the labels can be mapped (or normalized) to the range of $[0, 1]$. The discriminator is trained by using the combination of the usual unsupervised GAN loss function and a supervised regression loss:

$$\begin{aligned}&\text{Loss}_D = L_{\text{unsupervised}} + L_{\text{supervised}} \\ &L_{\text{unsupervised}} = \\ &E_{x\sim P_{data}(x)}[(1-D(x))^2] + E_{z\sim P_z(z)}[D(G(z))^2] \\ &L_{\text{supervised}} = \|y-\hat{y}\|\end{aligned} \tag{3}$$

where z is the noise drawn from a uniform or a normal distribution. x and $G(z)$ describe the true and generated images respectively. The term y refers to the true value of the label and $\hat{y}$ indicates the predicted labels. It is worth mentioning that we employed the least-square loss function introduced in

[20] in the unsupervised part of the equation. In addition, the supervised regression error (i.e. the difference between the predicted and the true labels) is added to the discriminator loss function which helps to generate labels for the unseen or generated samples. In the second approach (see Fig. 2), instead

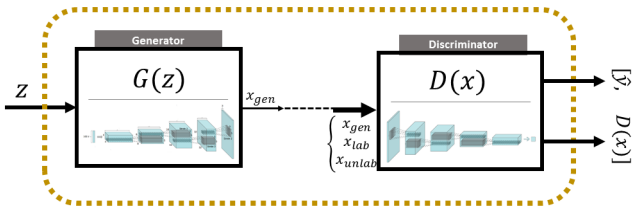

Fig. 1: Architecture 1: the proposed Reg-GAN where both $D(x)$ and the predicted labels are generated from the deep convolutional neural network. x_{gen}, x_{lab}, and x_{unlab} represent the unlabeled generated, labeled real and unlabeled real samples respectively.

of having two outputs in the discriminator, we keep only its regression output to predict labels. Then we feed the labels to another function to assign an index to the input based on the predicted label. In other words, instead of differentiating true and generated samples by the network directly, we can employ a separate kernel function (Eq. 4) on the regression output for deciding whether the predicted labels are realistic or not. The kernel function is responsible for assigning an index to the predicted label of each input. If the predicted label is within the normalized range of true labels, i.e. between 0 and 1, then the assigned index is 1. Otherwise, the index will be assigned a number exponentially smaller than 1 according to the distance of the predicted value from the target range of true labels. The training procedure of the proposed approaches are briefly portrayed in Algorithm 1.

$$\text{Kernel Function } K(\hat{y}) = \begin{cases} \exp(\hat{y}), & 0 \leq \hat{y} \\ 1, & 0 < \hat{y} \leq 1 \\ \exp((1-\hat{y})), & 1 < \hat{y} \end{cases} \quad (4)$$

5. EXPERIMENTS AND RESULTS

In this section, our method will be evaluated from different point of views such as regression prediction error and quality of generated samples. The main objective of the experiments is to show that our proposed architectures are able to learn data generation and label prediction even in the case of having limited labeled data.

Algorithm 1 Semi-supervised regression with GAN. We use default values for $\alpha = 0.0005$, β=0.5

Require: The Adam hyperparameters α, β, the number of batches m

Require: Initial discriminator parameters w_0 and initial generator parameters θ_0

while θ has not converged **do**
  for $i = 1, ..., m$ **do**
    Sample real data $x, y \sim P_{data}(x, y)$, z $\sim P_z(z)$
    $L_D^{(i)} \leftarrow E_{x\sim P_{data}(x)}[(1 - D(x))^2] + E_{z\sim P_z(z)}[D(G(z))^2] + \|y - \hat{y}\|$
    $w \leftarrow Adam(L_D^{(i)}, w, \alpha, \beta)$
    $L_G^{(i)} \leftarrow L_{\text{feature\_matching}}$
    $\theta \leftarrow Adam(L_G^{(i)}, \theta, \alpha, \beta)$
  end for
end while

5.1. Data and experimental setup

For our experiments, we use a publicly available driving dataset [1]. The dataset contains images taken from a front facing camera mounted on the car with their corresponding steering angles as labels. We randomly choose 7200 samples from the dataset for training and 9000 samples for test. We aim at evaluating our technique when the number of available samples and the number of labeled samples are small. Hence, we did not incorporate more samples from the test set into our training data. The label of the samples falls within the range of [-2.79, 8.75] which are further normalized into the range of [0, 1] in a linear way. We use the average normalized prediction error over the test set to compute the test error as following:

$$\text{test\_error} = \frac{1}{N}\sum_{j=1}^{N}\frac{||\hat{y_j} - y_j||}{||y_{max} - y_{min}||} \times 100 \quad (5)$$

where N is the number of test samples, and y_{min} and y_{max} represents the minimum and maximum value of the groundtruth labels (i.e. 0 and 1 respectively). We use the available Improved-GAN code, which is written using the 'Theano' Python library and the 'Lasagne' deep learning library, as a baseline to implement our proposed methods. We perform experiments for 800 iterations with a learning rate of α=0.0005 for our methods and 0.0003 for the original improved-GAN [1]. The experiments are run on a single NVIDIA Tesla P100 GPUs.

5.2. Experimental Results

For training the methods, different number of labeled samples (1000, 2000, 4000, and "All") from the training set are used in a semi-supervised learning setting. Moreover, for each experiment, all the images in the training set are fed to the algorithm

[1]The dataset can be downloaded from: `https://drive.google.com/file/d/0B-KJCaaF7elleG1RbzVPZWV4Tlk/view`

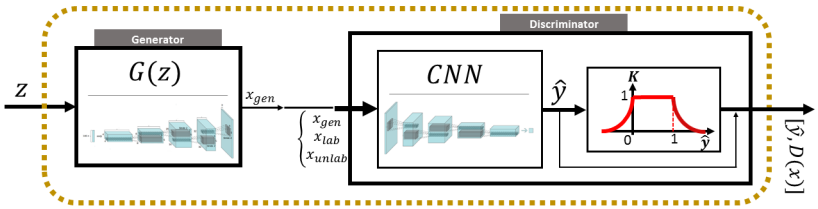

Fig. 2: Architecture 2: the proposed Reg-GAN where only the labels are predicted by a deep CNN.

(a) Architecture 1 (b) Architecture 2 (c) Improved-GAN

Fig. 3: Sample generated images by using (a) Architecture 1, (b) Architecture 2, and (c) Improved-GAN when 1000 labeled samples are used for training.

as unlabeled samples. We compare our proposed architectures with the state-of-the-art Improved-GAN semi-supervised learning approach [1]. We chose this method over other similar techniques because it outperforms them. In order to fit our dataset into the Improved-GAN classification framework, we discretized the normalized continuous labels into 10 number of classes (we assign labels in the range of [0, 0.1)→0, and [0.1 0.2)→ 1, and ... [0.9,1]→9). Bear in mind that this discretization will add some unavoidable quantization error to our training. The results of the experiments are shown in Table 1. From Table 1, we can note that our Reg-GAN outperforms the Improved-GAN approach in all scenarios significantly. Our method gives the average improvements of 42.7% and 29.7% over the traditional Improved-GAN approach for the architecture 1 and 2 respectively. An example of the generated samples from different techniques is shown in Fig. 3. These samples are derived after training the networks over 1000 samples.

Table 1: Test errors using 1000, 2000, 4000, and All labeled samples

Model	1000	2000	4000	All
Improved-GAN	4.38%	4.22%	4.07%	4.06%
Reg-GAN (Arch 1)	2.43%	2.40%	2.39%	2.36%
Reg-GAN (Arch 2)	3.81%	3.58%	2.23%	2.21%

6. CONCLUSION AND FUTURE WORK

This work concerned solving the semi-supervised learning regression task by incorporating generative adversarial networks. The conventional semi-supervised learning with GAN is suitable for classification tasks. Using them for the regression task requires to convert continuous labels to a limited number of classes. This conversion will add the quantization error to the training, and determining the number of classes for each application is non-trivial. This work proposes a semi-supervised regression task using GANs which overcomes the aforementioned problems that arise using semi-supervised classification techniques to solve the regression task. We did experiments on a publicly available driving dataset where continuous steering angles were used as the labels with the corresponding images. The experiments showed that our proposed approaches outperform the state-of-the-art Improved-GAN technique in the literature. We summarize our plan for future work in the following:

1. The idea of this work can be extended to cover classification problems as well by assigning the regression label output to predict the class labels. However, the performance of this approach on the classification problems needs to be investigated.

2. The idea of semi-supervised regression may have other applications such as face detection, and apparent age estimation from a single image. Our method can be evaluated on those applications as well.

7. REFERENCES

[1] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, "Improved techniques for training gans," *CoRR*, vol. abs/1606.03498, 2016.

[2] Alex Kuefler, Jeremy Morton, Tim Wheeler, and Mykel Kochenderfer, "Imitating driver behavior with generative adversarial networks," *arXiv preprint arXiv:1701.06699*, 2017.

[3] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al., "An empirical evaluation of deep learning on highway driving," *arXiv preprint arXiv:1504.01716*, 2015.

[4] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al., "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[5] Arna Ghosh, Biswarup Bhattacharya, and Somnath Basu Roy Chowdhury, "Sad-gan: Synthetic autonomous driving using generative adversarial networks," *arXiv preprint arXiv:1611.08788*, 2016.

[6] Ershad Banijamali, Ali Ghodsi, and Pascal Poupart, "Generative mixture of networks," *arXiv preprint arXiv:1702.03307*, 2017.

[7] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[8] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 791–798.

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

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[11] Martin Arjovsky and Léon Bottou, "Towards principled methods for training generative adversarial networks," in *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*, 2017, vol. 2016.

[12] Jost Tobias Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *arXiv preprint arXiv:1511.06390*, 2015.

[13] Augustus Odena, "Semi-supervised learning with generative adversarial networks," *arXiv preprint arXiv:1606.01583*, 2016.

[14] Chongxuan Li, Kun Xu, Jun Zhu, and Bo Zhang, "Triple generative adversarial nets," *arXiv preprint arXiv:1703.02291*, 2017.

[15] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589.

[16] Tan Nguyen, Wanjia Liu, Ethan Perez, Richard G Baraniuk, and Ankit B Patel, "Semi-supervised learning with the deep rendering mixture model," *arXiv preprint arXiv:1612.01942*, 2016.

[17] Ryosuke Tachibana, Takashi Matsubara, and Kuniaki Uehara, "Semi-supervised learning using adversarial networks," in *Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on*. IEEE, 2016, pp. 1–6.

[18] Emily Denton, Sam Gross, and Rob Fergus, "Semi-supervised learning with context-conditional generative adversarial networks," *arXiv preprint arXiv:1611.06430*, 2016.

[19] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko, "Semi-supervised learning with ladder networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 3546–3554.

[20] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley, "Least squares generative adversarial networks," *arXiv preprint ArXiv:1611.04076*, 2016.