Adaptive WGAN with loss change rate balancing

Xu Ouyang
Illinois Institute of Technology

Gady Agam
Illinois Institute of Technology

Abstract

Optimizing the discriminator in Generative Adversarial Networks (GANs) to completion in the inner training loop is computationally prohibitive, and on finite datasets would result in overfitting. To address this, a common update strategy is to alternate between k optimization steps for the discriminator D and one optimization step for the generator G. This strategy is repeated in various GAN algorithms where k is selected empirically. In this paper we show that this update strategy is not optimal in terms of accuracy and convergence speed, and propose a new update strategy for Wasserstein GANs (WGAN) and other GANs using the WGAN loss (e.g. WGAN-GP, Deblur GAN, and Super resolution GAN). The proposed update strategy is based on a loss change ratio comparison of G and D. We demonstrate that the proposed strategy improves both convergence speed and accuracy.

1. Introduction

GANs [8] provide an effective deep neural network framework that can capture data distribution. GANS are modeled as a min-max two-player game between a discriminator network \( D_\psi(x) \) and a generator network \( G_\theta(z) \). The optimization problem solved by GAN [21] is given by:

\[
\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{\text{data}}} [f(D(x))] + \mathbb{E}_{z \sim p_{\text{latent}}} [f(-D(G(z)))]
\]

where \( G : Z \to X \) maps from the latent space Z to the input space X; \( D : X \to \mathbb{R} \) maps from the input space to a classification of the example as fake or real; and \( f : \mathbb{R} \to \mathbb{R} \) is a concave function. In the remainder of this paper, we use the Wasserstein GAN [3] obtained when using \( f(x) = x \).

GANs have been shown to perform well in various image generation applications such as: deblurring images [17], increasing the resolution of images [18], generating captions from images [5], and generating images from captions [23]. Training GANs may be difficult due to stability and convergence issues. To understand this consider the fact that GANs minimize a probabilistic divergence between real and fake (generated by the generator) data distributions [22]. Arjovsky et al. [2] showed that this divergence may be discontinuous with respect to the parameters of the generator, and may have infinite values if the real data distribution and the fake data distribution do not match.

In order to solve a divergence continuous problem, WGAN [3] uses the Wasserstein-divergence by removing the sigmoid function in the last layer of the discriminator and so restricting the discriminator to Lipschitz continuous functions instead of the Jensen-Shannon divergence in the original GAN [8]. WGAN will always converge when the discriminator is trained until convergence. However, in practice, WGAN is trained with a fixed number (five) of discriminator update steps per generator update step.

Even though WGAN is more stable than the original GAN, Mescheder et al. [19] proved that WGAN trained with simultaneous or alternating gradient descent steps with a fixed number of discriminator updates per generator update and a fixed learning rate \( h > 0 \) does generally not converge to the Nash equilibrium for a Dirac-GAN, where
the Dirac-GAN \cite{mescheder2018training} is a simple prototypical GAN. Furthermore, in WGAN, the authors alternated between five optimization steps for the generator and one optimization step for the discriminator where there the ratio is set empirically. In this paper we address the number of training steps for the discriminator and generator and show how to adaptively change it during training so as to make the training converge faster to a more accurate solution.

The strategy we propose for balancing the training of the generator and discriminator is based on the discriminator and generator loss change ratios ($r_d$ and $r_g$, respectively). Instead of a fixed update strategy we decide whether to update the generator or discriminator by comparing the weighted loss change ratios $r_d$ and $\lambda \cdot r_g$ where the weight $\lambda$ is a hyper-parameter assigning importance to $r_g$. Using the weight $\lambda$ it is possible to give preference to the update of one of the components thus, for example, training the discriminator more frequently as in the original WGAN. The reason for giving preference to one component has to do with the fact that the individual components affect the divergence reduction at different rates. Following Mescheder et al. \cite{mescheder2018training} we demonstrate that the proposed strategy can reach a local convergence point for the Dirac-GAN problem, unlike the original update strategy which cannot achieve it.

To further demonstrate the advantage of the proposed update strategy, we train the WGAN\cite{goodfellow2014generative}, WGAN-GP\cite{gulrajani2017improved}, Deblur-GAN\cite{isola2017image}, and SR-WGAN\cite{bahng2019srgan} using the proposed strategy using different image datasets. These represent a wide range of GAN applications. Experimental results show that in general the proposed strategy converges faster while achieving in many cases better accuracy. An illustration is provided in Figure 1 where the proposed adaptive WGAN is compared with the traditional WGAN update strategy.

The main contribution of this paper is in proposing an adaptive update strategy for WGAN instead of the traditional fixed update strategy in which the update rate is set empirically. We show that the proposed strategy can reach local convergence for Dirac-GAN, unlike the traditional fixed update strategy which cannot do so. Experimental results on several problems with several datasets show that the proposed adaptive update strategy results in faster convergence and/or higher performance.

2. Related work

The question of which training methods for GANs actually converge was investigated by Mescheder et al. \cite{mescheder2018training} where they introduce the Dirac-GAN. The Dirac-GAN consists of a generator distribution $p_G = \delta_0$ and a linear discriminator $D_\theta(x) = \psi \cdot x$. In their paper they prove that a fixed point iteration $F(x)$ is locally convergent to $x$, when the absolute values of the eigenvalues of the Jacobian $F'(x)$ are all smaller than 1. They further prove that for the Dirac-GAN, with both simultaneous and alternative gradient descent updates, the absolute values of the eigenvalues of the Jacobian $F'(x)$ in GANs with unregularized gradient descent (which include the original GAN, WGAN and WGAN-GP) are all larger or equal to 1, thus showing that these types of GANs are not necessarily locally convergent for the Dirac-GAN. To address this convergence issue, Mescheder et al. \cite{mescheder2018training} added gradient penalties to the GAN loss and proved that regularized GAN with these gradient penalties can reach local convergence. This solution does not apply to WGAN and WGAN-GP which remain not locally convergent problems. Note that the WGAN is generally more stable and easier to train compared with GAN and hence the need for the adaptive update scheme we propose in this paper.

Heusel et al. \cite{heusel2017gans} attempt to address the convergence problem in a different way by altering the learning rate. In their approach they use a two time-scale update rule (TTUR) for training GANs with stochastic gradient descent using arbitrary GAN loss functions. Instead of empirically setting the same learning rate for both the generator and discriminator, TTUR uses different learning rates for them. This is done in order to address the problem of slow learning for regularized discriminators. They prove that training GANs with TTUR can converge to a stationary local Nash equilibrium under mild condition based on stochastic approximation theory. In their experiments on image generation, the show that WGAN-GP with TTUR gets better performance. Note however that empirically setting the learning rate is generally difficult and even more so when having to set two learning rates jointly. This makes applying this solution more difficult.

It is well understood that the complexity of the generator should be higher than that of the discriminator, a fact that makes GANs harder to train. Balancing the learning speed of the generator and discriminator is a fundamental problem. Unbalanced GANs \cite{gulrajani2017improved} attempt to address this by pre-training the generator using variational autoencoder (VAE \cite{kingma2013auto}), and using this pretrained generator to initialize the GAN weights during GAN training. An alternative solution is proposed in BEGAN\cite{mathieu2016unbalanced}, where the authors introduce an equilibrium hyperparameter ($E[f(-D(G(z)))] / E[f(D(x))]$) to maintain the balance between the generator and discriminator. Training the two neural networks in this approach is time consuming and the equilibrium hyper-parameter is not suitable for all GAN training cases. For example it is not suitable when there is a content loss in the generator loss term $L_G$ as $L_G$ and the discriminator loss $L_D$ are not on the same scale.

A similar issue to the unbalanced training of the generator and discriminator in GANs arises in imbalanced training of multiple task networks. The GradNorm \cite{zhang2018gradnorm} approach provides a solution to balancing multitask network train-
ing based on gradient magnitudes. In this approach, the authors multiply each of the single-task loss terms by weights, and automatically update those weights by computing a gradient normalization term. A relative inverse training rate \((L_{\text{current}}/L_{\text{initial}})\) for each task is used to compute this normalization term. This strategy depends on a common loss term minimization where individual task terms are weighted and so is not suitable for GANs where there is no shared layer as in multi-task networks.

3. Method

3.1. WGAN update strategy

In this section, we present our proposed update strategy to automatically set the update rate of the generator and discriminator instead of using a fixed rate as is commonly done. In WGAN or any GAN based on the WGAN loss, the Nash equilibrium is reached when the generator and discriminator loss terms stop changing. That is:

\[|L_{\text{g}}^c - L_{\text{d}}^c| = 0 \quad \& \quad |L_{\text{g}}^d - L_{\text{d}}^d| = 0 \]  \hfill (2)

where \(L_{\text{g}}^c, L_{\text{d}}^c\) represent the generator and discriminator loss in the current iteration respectively, and \(L_{\text{g}}^p, L_{\text{d}}^p\) represent the respective loss terms in the previous iteration. Since we play a min-max game in WGAN, it is crucial to balance the respective loss terms in the previous iteration. Since we cannot prioritize the update of one component over the other as commonly done in GANs, we use an importance parameter \(\lambda\). Thus, if \(r_d > \lambda \cdot r_g\), we update the discriminator, or otherwise update the generator. A larger loss change ratio of one component means that this component is in greater need for update. The details of our proposed adaptive WGAN are provided in Algorithm 1.

3.2. Convergence analysis

In this section, we demonstrate that with our proposed update strategy, WGAN can reach local convergence for Dirac-GAN, whereas it cannot do so with a fixed update strategy. The GAN objective function is given by:

\[L(\theta, \psi) = E_{p(x)}[f(D_\psi(x))] + E_{p(z)}[f(-D_\psi(G_\theta(z)))] \]  \hfill (5)

The discriminator attempts to maximize this function whereas the generator attempts to minimize it. The goal is to find a Nash-equilibrium, where both components cannot improve their utility. The optimization is normally done using an alternating gradient descent where when training the generator the parameters are updated by:

\[\theta_{t+1} = \theta_t + \alpha \cdot v(\theta_t)\]

and when training the discriminator the parameters are updated by:

\[\psi_{t+1} = \psi_t + \alpha \cdot v(\psi_t)\]

\[\theta_{t+1} = \theta_t\]

The Dirac-GAN [19] consists of a generator distribution \(p_\theta = \delta_0\) and a linear discriminator \(D_\psi(x) = \psi \cdot x\). The true

\begin{algorithm}

\textbf{Proposed adaptive WGAN}

\begin{itemize}
  \item \textbf{parameters}: learning rate (\(\alpha\)); clipping parameter (\(c\)); loss importance (\(\lambda\)); batch size (\(m\)).
  \item \textbf{variables}: generator parameters (\(\theta\)); discriminator parameters (\(\psi\)); generator loss change ratio (\(r_g\)); discriminator loss change ratio (\(r_d\)). The loss change ratios are initialized to 1.
\end{itemize}

\begin{algorithmic}
  \State \textbf{while} \(\theta\) has not converged \textbf{do}
  \State \hspace{1em} Sample \(\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r\) a batch from the real data
  \State \hspace{1em} Sample \(\{z^{(i)}\}_{i=1}^m \sim p(z)\) a batch of prior samples
  \State \hspace{1em} if \(r_d > \lambda \cdot r_g\) then \# update the discriminator
  \State \hspace{2em} \(g_{\psi} \leftarrow \nabla_{\psi} \frac{1}{m} \sum_{i=1}^m f_\psi(g_\theta(z^{(i)})) - \frac{1}{m} \sum_{i=1}^m f_\psi(x^{(i)})\)
  \State \hspace{2em} \(\psi \leftarrow \psi + \alpha \cdot \text{RMSProp}(\psi, g_\theta)\)
  \State \hspace{2em} \(\psi \leftarrow \text{clip}(\psi, -c, c)\) \# update the generator
  \State \hspace{2em} \(g_\theta \leftarrow -\nabla_{\theta} \frac{1}{m} \sum_{i=1}^m f_\psi(g_\theta(z^{(i)}))\)
  \State \hspace{1em} \(\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)\)
  \State \hspace{1em} \textbf{end if}
  \State \hspace{1em} if first iteration then \(L_{\text{g}}^p, L_{\text{d}}^p = L_{\text{g}}, L_{\text{d}}\)
  \State \hspace{1em} \textbf{end if}
  \State \hspace{1em} \(L_{\text{g}}^c, L_{\text{d}}^c = L_{\text{g}}, L_{\text{d}}\)
  \State \hspace{1em} \(r_g, r_d = (|L_{\text{g}}^c - L_{\text{d}}^c|)/L_{\text{g}}, (|L_{\text{d}}^e - L_{\text{d}}^d|)/L_{\text{d}}^d\)
  \State \hspace{1em} \(L_{\text{g}}^p, L_{\text{d}}^p = L_{\text{g}}, L_{\text{d}}\)
  \State \hspace{1em} \textbf{end while}
\end{algorithmic}

\end{algorithm}

\[\text{Algorithm 1 Proposed adaptive WGAN} \]

\[\text{parameters: learning rate (\(\alpha\)); clipping parameter (\(c\)); loss importance (\(\lambda\)); batch size (\(m\)).}
\[\text{variables: generator parameters (\(\theta\)); discriminator parameters (\(\psi\)); generator loss change ratio (\(r_g\)); discriminator loss change ratio (\(r_d\)). The loss change ratios are initialized to 1.}
\]
data distribution $p_D$ is given by a Dirac-distribution concentrated at 1. Thus, there is one parameter $\theta$ in the generator and one parameter $\psi$ in the discriminator. For WGAN, we define $f(t) = t$ and add a Lipschitz constraint (-0.5, 0.5) on the discriminator as in the original WGAN. Thus, the GAN objective function in Equation 5 is given by:

$$L(\theta, \psi) = \psi \cdot 1 - \psi \cdot \theta$$

(8)

The unique equilibrium point of the objective function in Equation 5 is $\theta = 1, \psi = 0$. Since $\psi(\theta, \psi) = 0$ if and only if $(\theta, \psi) = (1, 0)$ as shown by:

$$\psi(\theta, \psi) = \left( \begin{array}{c} \psi \\ 1 - \theta \end{array} \right)$$

(9)

Thus, when training the generator, the parameters update in Equation 6 are given by:

$$\begin{pmatrix} \theta_{t+1} \\ \psi_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & \alpha \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_t \\ \psi_t \end{pmatrix}$$

(10)

When training the discriminator, the parameters update in Equation 7 are given by:

$$\begin{pmatrix} \theta_{t+1} \\ \psi_{t+1} \end{pmatrix} = \begin{pmatrix} 0 & -\alpha + \frac{\alpha}{\beta} \\ 1 - \alpha + \frac{\alpha}{\beta} \end{pmatrix} \begin{pmatrix} \theta_t \\ \psi_t \end{pmatrix}$$

(11)

We employ our proposed update strategy using an alternating gradient descent based on Equations 10 and 11. We decide on the component to update by comparing the loss change ratios ($r_d$ and $r_g$) as described in Algorithm 1. The importance parameter $\lambda$ is set to 1. For comparison we also apply the original WGAN update strategy of alternating gradient descent with fixed update steps (5 discriminator updates for each generator update). The results are shown in Figure 2. As can be observed in sub-figure (a) fixed WGAN updates ($n_d = 5, n_g = 1$) do not converge whereas in sub-figure (b) adaptive updates following the proposed approach do converge to the Nash equilibrium point (1, 0).

3.3. Network architectures

To demonstrate our proposed adaptive training strategy as described in Algorithm 1 we evaluated several network architectures with and without adaptive training. Specifically, we evaluated standard WGAN [3] and WGAN-GP [9] networks for image synthesis, Deblur GAN [17] for image debluring using a Conditional Adversarial Network [20], and Super Resolution WGAN [13] for increasing image resolution using perceptual loss, content loss, and WGAN loss. Except for modifying the update strategy to become adaptive, we retained the original optimizer and loss functions in each of the evaluated networks, as can be found in the papers referenced above.

In our update strategy we set the importance parameter $\lambda$ to different values. We observe that higher values of $\lambda$ (up to 1000) perform better when the generation task is complex (e.g. in Deblur GAN and Super Resolution WGAN). Smaller values of $\lambda$ (e.g. 1-5) work for the WGAN and WGAN-GP image generation networks. Increasing the value of $\lambda$ results in training more the generator which is necessary due to the increased complexity of the generator. Note that training the generator more is in contrast to the suggestion in the original WGAN [3] paper where it is suggested to perform 5 training steps for the discriminator for each step of the generator. Experimental evaluation results are provided in the next section.

4. Experimental evaluation

In this section, we train different GANs with our updating strategy: WGAN, WGAN-GP, TTUR, Gradient Penalty, Deblur GAN and Super Resolution WGAN, and evaluate them both in quantitative and qualitative ways.

4.1. Experimental setup

We train both WGAN and adaptive WGAN for 500 epochs and using RmsProp optimizer (learning rate=0.00005 for both G and D). We train WGAN-GP and adaptive WGAN-GP for 20k iterations and using Adam optimizer (learning rate=0.0001 for both G and D, $\beta_1 = 0.5, \beta_2 = 0.9$). And both in original WGAN and WGAN-GP training, we follow the setting in [3] [9] that updates five times for D per updating one time for G. In adaptive WGAN and adaptive WGAN-GP training, we tried different $\lambda$: 1,3,5,10.

Meanwhile, we compare with the WGAN-GP TTUR (learning rate of G: $lr_g = 0.0001$, learning rate of D: $lr_d = 0.0003$ in TTUR; in adaptive WGAN-GP, $lr_g = 0.0003$ and $lr_d = 0.0003$ in order to keep in step), Ubalanced GAN, Gradient Penalty (we follow the hyper-parameters setting in original Gradient Penalty) and with TTUR ($lr_g = 0.0001$ and $lr_d = 0.0003$) and with our strategy ($lr_g = 0.0003$ and $lr_d = 0.0003$ in order to keep in step).
We train both Deblur GAN and adaptive Deblur GAN for 1500 epochs and using Adam optimizer (learning rate=0.0001 for both G and D, $\beta_1 = 0.5, \beta_2 = 0.999$). We train Deblur GAN 5 time on D and 1 time on G in each batch which followed the [17]. In adaptive Deblur GAN training, we tried $\lambda$: 1, 10, 100.

We train both Super Resolution WGAN and adaptive Super Resolution WGAN for 1000 epochs and using RmsProp optimizer (learning rate=0.001 for both G and D). We train Super Resolution WGAN one time on D and one time on G in each batch which followed the [13]. In adaptive Super Resolution WGAN training, we tried $\lambda$: 1, 10, 100, 1000.

### 4.2. Datasets

To train the WGAN we use a 100 dimensional random noise vectors as input. For targets we use 3 datasets: the CIFAR-10 [16] which includes 50,000 training examples and 10,000 validation examples; the LSUN [28] conference room dataset which has 229,069 training examples and 300 validation examples; and the labeled faces in the wild (LFW [12]) dataset which has 13,233 examples split into 10587 training examples, 1323 validation examples, and 1323 testing examples. For all of three datasets, we set up the image size to 64x64 to match the original paper [3] setting.

To train the WGAN-GP and compare with the TTUR, Gradient Penalty, we all use the 128 dimensional random noise vectors as the input, and the CIFAR-10 dataset as the targets where the images are resized to 32x32 to match the original paper [9] setting.

To train the Deblur GAN we use the Caltech-UCSD Birds-200-2011 [26] dataset which has 200 classes of bird images with size 256x256. We synthesize blurred images from the original images using a sequence of six 3x3 Gaussian kernel convolutions. We then use the synthetic blurred images as the inputs and the corresponding original images as targets.

To train the Super Resolution WGAN (SR-WGAN) we use the DIV2K [1] dataset containing a diverse set of RGB images. In this set there are 700 training images, 100 validation images, and 100 test images. The images in this dataset are of various sizes. To synthesize the source data
we downscale each image by a factor of two, 4 times thus resulting in images having 1/16 size in each spatial dimension. The original images are then used as the corresponding targets.

4.3. Metrics

For evaluating image synthesis by WGAN and WGAN-GP we use as evaluation metrics the Inception Score (IS) \[^{[25]}\] and Frechet Inception Distance (FID) \[^{[7]}\] which are well known, commonly used, GAN performance evaluation metrics. The Inception Score can measure a synthetic image quality by computing the expected Kullback-Leibler divergence (KL divergence) between the marginal class distribution and conditional label distribution:

\[
IS = \exp(\mathbb{E}_x KL(p(y|x)||p(y)))
\]

(12)

where \(p(y|x)\) is the conditional label distribution of features extracted from the middle layers of the pretrained Inception-v3 model for generated images, and \(p(y)\) is the marginal class distribution. A higher IS value indicates better quality.

The Frechet Inception Distance \[^{[7]}\] that is given by:

\[
d^2(F,G) = |\mu_x - \mu_y|^2 + tr[\Sigma_x + \Sigma_y - 2(\Sigma_x \Sigma_y)^{1/2}]\]

(13)

where \(F, G\) are two distributions of features extracted from the middle layers of a pretrained Inception-v3 model for generated and real images. The parameters \(\mu_x, \mu_y, \Sigma_x, \Sigma_y,\) are the mean vectors and covariance matrices of \(F\) and \(G\). A lower FID score (the distribution \(F\) is similar to the distribution \(G\)) indicates better quality results. During training, in order to assess the convergence speed, we recorded each epoch’s FID and saved the models in which epoch had the lowest FID value.

To evaluate the deblur and super resolution networks, we use two common metrics to measure the similarity between generated (deblurred or increased resolution) images and the corresponding target images: structural similarity index (SSIM) \[^{[27]}\], and peak signal to noise ratio (PSNR). The SSIM measure is based on computing the mean, variance and covariance of a variety of windows in the compared images. The PSNR is based on the inverse of the mean squared error (MSE). Generally, for both measures, higher values indicate the generated image is more similar to the target image. During training steps we recorded the PSNR in each epoch to evaluate the convergence speed and saved the best PSNR performance models.

4.4. Qualitative evaluation

Figure 3 shows generated images with WGAN and WGAN-GP trained on the CIFAR-10 dataset using a fixed training strategy and the proposed adaptive training strategy. As can be observed, the proposed adaptive WGAN training strategy progresses faster than a fixed training strategy. When comparing the best epoch (the epoch with the best FID) results, we observe that the adaptive WGAN results are more realistic compared with the original WGAN results. Both WGAN-GP and the adaptive WGAN-GP can get some meaningful results, but the adaptive strategy progresses faster. Figure 4 shows images generated with WGAN and the proposed adaptive WGAN trained on the LSUN conference rooms and the LFW datasets. We observe that the adaptive WGAN results have more details and are more realistic on both datasets.

4.5. Quantitative evaluation

To evaluate WGAN and WGAN-GP for image synthesis we use the Inception Score (IS) and the Frechet Inception Distance (FID). We train the networks using both the fixed strategy and the proposed adaptive strategy (with different importance parameter \(\lambda\) values) on the CIFAR-10 dataset and record the first epoch when a target FID value is obtained. We record in addition the total number of \(G\) and \(D\) updates. The results are shown in Table 1. As can be observed the proposed adaptive training scheme converges faster and to a better result when \(\lambda\) is between 1 and 5 (with best result at \(\lambda = 3\)).

When considering the total number of \(G\) and \(D\) updates, we see that with \(\lambda = 3\) the proposed adaptive training strategy trains \(G\) three to four times more than \(D\), whereas the
suggested ratio in the fixed training scheme [8] [24] [3] is to train D five times more than G. The update frequency of G and D can be effected by multiple factors such as the complexity of the models, the optimizer and its parameters (e.g. learning rate), and the loss function. It is therefore difficult to empirically estimate the G and D training ratio. Moreover, even if the training ratio of G and D is somehow determined (e.g. hyper-parameter search) it may change during iterations as the algorithm gets close to convergence. The proposed adaptive training strategy alleviates the need to carefully set this parameter and provides a systematic way to continuously estimate it. While the proposed adaptive scheme still involves selecting an importance parameter λ, training results are less sensitive to the selection of this parameter and a simple default (e.g. λ = 1) may suffice.

A similar evaluation of the proposed adaptive WGAN training strategy when trained with different datasets is provided in Table 2. In this table training is done separately both with the LSUN conference room and LFW datasets. The evaluation on these datasets results in similar conclusions to the ones obtained when training with the CIFAR-10 dataset. The proposed adaptive training strategy converges faster and to a better result.

Comparison of various WGAN training schemes targeting balancing G and D training (see Section 2). Methods with “adaptive” in their name employ the proposed adaptive training scheme. The network was trained using the CIFAR-10 dataset (batch-size=64, imagesize=64x64, 500 epochs). The symbol ‘*’ indicates that no data is available from the original paper. As can be observed, while the fixed training strategy gives preference to training D, the proposed adaptive strategy ends up giving preference to training G and by doing so converges faster and to a better result. The parameters n_g, n_d is the set number of fixed update steps. A ‘*’ indicates the score could not be reached by the training strategy.

Table 1. Comparison of fixed WGAN update strategy to the proposed adaptive WGAN update strategy with different importance coefficients λ trained on the CIFAR10 dataset (batchsize=64, image-size=64x64, 500 epochs). Columns 3-6 show the first epoch that reached the target FID value. Columns 7-8 show the result of the best epoch up to 500. The last to columns show the actual number of updates of G and D at the best epoch. As can be observed, the proposed adaptive training strategy converges faster and to a better result. The parameters n_g, n_d is the set number of fixed update steps. A ‘*’ indicates the score could not be reached by the training strategy.

Table 2. Comparison of fixed WGAN update strategy to the proposed adaptive WGAN update strategy with different importance coefficients λ trained on the LSUN conference room and LFW datasets (using the same experiment setting as in Table 1). The proposed training strategy converges faster and to a better result.

Table 3. Comparison of various WGAN training schemes targeting balancing G and D training (see Section 2). Methods with “adaptive” in their name employ the proposed adaptive training scheme. The network was trained using the CIFAR-10 dataset (batch-size=64, imagesize=32x32, 1000 epochs). The symbol ‘*’ indicates that no data is available from the original paper. As can be observed, the proposed adaptive training scheme converges faster.
Table 4. Comparison of Deblur GAN trained without and with the proposed adaptive training scheme. The network was trained using the CUB-200-2011 bird dataset. The importance parameter $\lambda$ in the proposed adaptive update strategy is attempted with different values. Columns 3-5 show the first epoch that reached the target PSNR value. The last two columns show the total number of training epochs for D and G. As can be observed the proposed adaptive training scheme trains the generator more than the discriminator whereas when employing the suggested fixed strategy trains the discriminator more than the generator. We observe that the proposed adaptive training strategy (with all $\lambda$ values) converges faster and to a better result.

Table 5. Comparison of the super resolution WGAN (SR-WGAN) trained without and with the proposed adaptive training scheme. The network was trained using the DIV2K dataset. The importance parameter $\lambda$ in the proposed adaptive update strategy is attempted with different values. Columns 3-5 show the first epoch that reached the target PSNR value. The last two columns show the total number of training epochs for D and G. We observe that the proposed adaptive training strategy (with $\lambda = 1000$) converges faster and to a better result.

10 dataset. Methods with “adaptive” in their name employ the proposed adaptive training scheme. As can be observed, the proposed adaptive training scheme converges faster.

The deblur GAN and super resolution WGAN networks are evaluated in Tables 4 and 5 respectively. The Deblur GAN network is trained using the Caltech-UCSD Birds-200-2011 dataset whereas the SR-WGAN network is trained on the DIV2K dataset. Training in both cases is done both with a fixed update strategy and the proposed adaptive training strategy. Evaluation is done using SSIM and PSNR. In the evaluation we record the first epoch during training where a target PSNR value is achieved. In addition we record the total number of update steps for the generator and the discriminator. We observe that the proposed adaptive training scheme converges faster and to a better result for both the deblur and super resolution networks. Further, here too the ratio of training steps for G and D obtained by the proposed adaptive training strategy does not match the recommended ratio of training steps thus supporting the need for the proposed adaptive training scheme.

5. Conclusion

In this paper we propose an adaptive WGAN training strategy which automatically determines the sequence of generator and discriminator training steps. The proposed approach compares the loss change ratio of the generator and discriminator to decide on the next component (G or D) to be trained and so balances the training rate of the generator and discriminator. We show that a WGAN with this strategy could reach the local Nash Equilibrium point for the Dirac-GAN. Experimental evaluation results using different networks and datasets show that the proposed adaptive training scheme normally converges faster and to a lower minimum. Another advantage of the proposed adaptive update strategy is that it alleviates the need to empirically determine the number of update steps for the generator and discriminator. In future work, we will investigate additional update strategies suitable for various GAN structures and loss terms.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, July 2017.
[2] Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In 5th International Conference on Learning Representations, ICLR 2017, 2017.
[3] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017.
volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 2017.

[4] David Berthelot, Tom Schumm, and Luke Metz. Began: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717, 2017.

[5] Chen Chen, Shuai Mu, Wanpeng Xiao, Zexiong Ye, Liesi Wu, and Qi Ju. Improving image captioning with conditional generative adversarial nets. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 8142–8150, 2019.

[6] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 793–802. PMLR, 2018.

[7] D. C. Dowson and B. V. Landau. The Frechet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 123(3):450–455, September 1982.

[8] 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 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5767–5777, 2017.

[10] Hyungrok Ham, Tae Joon Jun, and Daeyeong Kim. Unbalanced gans: Pre-training the generator of generative adversarial network using variational autoencoder. *CoRR*, abs/2002.02112, 2020.

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 6626–6637, 2017.

[12] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[13] JustinhoCHN. Srgan wasserstein. GitHub repository, 2018.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[15] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012*, pages 1106–1114, 2012.

[17] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiri Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In 2018 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pages 8183–8192, 2018.

[18] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pages 105–114, 2017.

[19] Lars M. Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3478–3487, 2018.

[20] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[21] Vaishnavh Nagarajan and J. Zico Kolter. Gradient descent gan optimization is locally stable. In *Advances in Neural Information Processing Systems 30*, pages 5585–5595, 2017.

[22] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan:training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 271–279, 2016.

[23] Xu Ouyang, Xi Zhang, Di Ma, and Gady Agam. Generating image sequence from description with LSTM conditional GAN. In *24th International Conference on Pattern Recognition, ICPR 2018*, pages 2456–2461, 2018.

[24] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[25] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 2226–2234, 2016.

[26] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

[27] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

[28] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015.