Multiplicative Noise Channel in Generative Adversarial Networks

Xinhan Di
Deepearthgo
Deepearthgo@gmail.com

Pengqian Yu
National University of Singapore
yupengqian@u.nus.edu

Abstract

Additive Gaussian noise is widely used in generative adversarial networks (GANs). It is shown that the convergence speed is increased through the application of the additive Gaussian noise. However, the performance such as the visual quality of generated samples and semi-classification accuracy is not improved. This is partially due to the high uncertainty introduced by the additive noise. In this paper, we introduce multiplicative noise which has lower uncertainty under technical conditions, and it improves the performance of GANs. To demonstrate its practical use, two experiments including unsupervised human face generation and semi-classification tasks are conducted. The results show that it improves the state-of-art semi-classification accuracy on three benchmarks including CIFAR-10, SVHN and MNIST, as well as the visual quality and variety of generated samples on GANs with the additive Gaussian noise.

1. Introduction

Deep generative networks, especially the generative adversarial networks (GANs) [8], demonstrate the ability of generating real visual samples from a latent probabilistic space. A family of GAN architectures have been developed including DCGAN [17], ImprovedGAN [20], WGAN [1] and etc. The GANs networks are applied widely into a range of different tasks including supervised-classification [19], 3D reconstruction [29], representation of video [24], cross-domain transformation [14], sequence learning [16], image quality boosting [22] and other tasks.

Improvement for better convergence, achievement for higher quality of generated samples and a larger variety of visual patterns are the main research topics of GANs. For example, the inverse mapping function of the generator is estimated in an auto-encoder architecture and a two-way mapping of the generation is then calculated [13, 18]. The visual quality is shown to be improved. Different formulation of the distance are applied such as Wasserstein distance [1], least squares loss distance [15] and the adaptive hinge distance [26] are applied in the generator/discriminator. Both the visual quality and visual variety are shown to be improved. Multiple number of generators/discriminators are also applied to improve the visual quality of generated samples, including an array of discriminators applied in the GANs [6], message-sharing mechanism developed for multiple generators [7], triple-GAN designed for three-player formulation under GANs framework [12].

The introduction of noise in deep neural networks has been widely investigated. In particular, the additive noise is added to the input [30], the output [3], the latent vectors [20] and the gradient of deep neural networks [27]. Additive noise is shown to excite the speed of the convergence of training [3]. Moreover, among many techniques that are already developed for the improvement of GANs, additive noise such as additive Gaussian noise is widely used to improve the convergence the GANs [19]. However, to our best knowledge, additive noise is not helpful for the improvement of visual quality and variety through unsupervised learning of features. In this paper, instead of applying additive noise, a different form of noise called multiplicative noise is proposed and studied for the unsupervised learning of features.

In this work, we show that the introduction of multiplicative noise enjoys lower uncertainty than the additive noise under mild assumptions. Moreover, it is further analyzed that the multiplicative noise has lighter effects of mixing data than the additive noise under the GANs framework. We further introduce simple multiplicative noise to improve the accuracy of semi-classification, visual quality and visual variety of generated samples in comparison with the additive noise. We believe our current work is only an initial study of noise (rather than additive noise) in unsupervised deep neural networks. As is well known, the cortex is found to propagate noisy code [21]. Specifically, noise substantially increases the information capacity of the neuronal population [4]. We demonstrate that the multiplicative noise, even in a very simple formulation, obtains advantages than the additive noise which is widely used in deep learning networks. Moreover, we further study applicable formulations of noise which are very likely to improve the performance

of deep neural networks.

The contribution of our current work is listed as follows. First, we introduce multiplicative noise that has a very simple form and demonstrate its use in unsupervised deep neural networks. The noise can be shown to have with less uncertainty than the additive noise under mild conditions. Second, we show its practical use in deep neural networks. Particularly, we use extensive numerical experiments for semi-supervised classification. In comparison with the additive noise in the GANs, the multiplicative noise is shown to improve the semi-classification accuracy by about 2.16%, 1.85% and 0.11% for three benchmark datasets including SVHN,CIFAR-10 and MNIST, respectively when the base model is Improve-GAN [19] with additive noise. It achieves the state-of-art of semi-classification 4.19% and 11.91% for SVHN and CIFAR-10 respectively when the based model is π-model with additive Gaussian noise. Furthermore, both visual quality and visual variety of generated samples are improved when the based models are GAN+AN (additive noise) [17] and WGAN+AN (additive noise) [1], respectively.

1.1. Related Work

The insertion of noise in deep neural networks has been widely investigated. In particular, the additive noise is added to the input [30], the output [3], the latent vectors [20] and the gradient of deep neural networks [27]. For example, weights decay [9] is added as additional constraints with side-effect that each weight has unexpected magnitude, and the uncertainty of weights is measured [5]. Its practicality is limited since the weight is assumed to follow a diagonal Gaussian distribution. Slow convergence and even oscillation may occur with any unsuitable noise injection.

Special additive noise is shown to excite the speed of the convergence of training [3]. Since the injected noise in the output speeds up convergence of the EM algorithm on average [2], noisy convolutional neural network algorithm is developed [3, 25]. However, extra loss function is used for the additive noise. Therefore, there is a trade-off between the final accuracy and the extra loss [28]. Although the proposed multiplicative noise has a simple formulation, it is demonstrated to introduce less uncertainty into the deep neural network system than the additive noise. In practice, it is shown to improve the performance of GANs such as the improvement of semi-classification accuracy and visual quality without the extra regularization term.

Deep generative networks, especially the GANs [8], demonstrate the ability of generating real visual samples from a latent probabilistic space. A family of GAN architectures have been developed including DCGAN [17], WGAN [1] and ImprovedGAN [20]. One common technique used for these GANs is the additive Gaussian noise. This noise is applied for fast convergence of training the GANs. However, no strong evidence is provided that the additive Gaussian noise is capable of improving the visual quality and variety of generated samples. In this paper, the application of the proposed multiplicative noise is shown to improve both the visual quality and variety of the generated samples through unsupervised learning of the features.

2. Preliminaries

The introduction of noise into the deep neural network is considered as the import of uncertainty into the networks. A large amount of uncertainty is considered as an negative effect to the performance of the networks. Therefore, a good measure of the introduced uncertainty is selected. It is then used to calculate the amount of uncertainty that is introduced into the network. Finally, a simple formulation of proposed multiplicative noise is provided.

2.1. Background

A multi-layer generative neural network is given and its generic function is denoted as F. The additive noisy channel is widely used in the network [30, 3, 27, 28]. Let $\mathbf{Y} = \mathbf{X} + \mathbf{Z}$, where $\mathbf{X}, \mathbf{Z}, \mathbf{Y} \in \mathbb{R}^N$. $\mathbf{X}$ can be the input/latent/output vectors of the network F. We further define the i^{th} element of $\mathbf{X}$ by $\mathbf{X}_i$.

Additive noise $\mathbf{Z}$ is widely applied in the network to speed up the convergence of training [3], where $\mathbf{Z}$ is independent with $\mathbf{X}$. The insertion of additive noise brings the uncertainty, which can be measured via the channel capability. Let $p_i(x)$ be the distribution of $\mathbf{X}_i$ and $\mathbf{Y}_i = \mathbf{X}_i + \mathbf{Z}_i$, where $\mathbf{X}_i$ and $\mathbf{Y}_i$ are the input and output of the noisy channel, respectively. Following [23], the channel capability C_i can be expressed as

$$C_i = \max_{p_i(x) \text{ s.t. } \mathbb{E}(\mathbf{X}_i^2) \leq \text{Pw}} I(\mathbf{X}_i; \mathbf{Y}_i), \quad \forall i = \{1, \ldots, N\},$$

where Pw is the maximum channel power, and $\mathbf{I}(X, Y)$ represents the mutual information of $\mathbf{X}$ and $\mathbf{Y}$.

2.2. Multiplicative Noise

Multiplicative noise is well known as the noise that is relative with the state of a system rather than non-relative with the state. In this setting, $\mathbf{Z}_i$ can be seen as the additive noise term for the training of deep neural networks, and $\mathbf{X}_i$ can be seen as the state which is determined via the training data for $i = \{1, \ldots, N\}$. Let $\mathbf{Z}_i^{'}$ denote a random variable representing a simple multiplicative noise. $\mathbf{Z}_i^{'}$ is co-related with $\mathbf{Z}_i$ and $\mathbf{X}_i$. Moreover, $P(\mathbf{Z}_i^{'}) = P(\mathbf{X}_i)P(\mathbf{Z}_i)$ and the output of the multiplicative noise channel is changed to $\mathbf{Y}_i = \mathbf{X}_i + \mathbf{Z}_i^{'}$.

3. Lower Uncertainty of the Multiplicative Noise

In this section, we show that the uncertainty of the additive noise channel is larger than the uncertainty of the proposed multiplicative noise channel. This result is given via the calculation of the channel capability of these two channels under mild conditions. Since low capability holds low uncertainty, the channel capability can be seen as a measure of the uncertainty when the noise is introduced.

3.1. Assumptions

First, we make an assumption about the probability distribution of $\mathbf{X}_i$.

Assumption 1 *$\boldsymbol{X}_i \sim p_i(x)$ where $\mathbb{E}(\boldsymbol{X}_i^2) \leq Pw \leq 1$ and $\mathbb{E}(\boldsymbol{X}_i) \in [-1, 1]$, $\forall i = \{1, \ldots, N\}$.*

This assumption is widely satisfied in the generative adversarial neural networks. In the training of the generative adversarial neural networks, the generated outputs are normalized and the value of each elements of the outputs are between -1 and 1. For example, the value of pixels of the generated 2D images is between 0 and 1. In addition, the value of the output of the discriminator is expected to range from 0 to 1. Therefore, the $\mathbf{X}_i,\ i = \{1, \ldots, N\}$, can be the generated outputs or the value of the discriminator, and the preceding assumption can be satisfied.

Second, another assumption about $\mathbf{Z}_i$ is given below.

Assumption 2 $\mathbb{E}(\mathbf{Z}_i) \geq 0,\ \forall i = \{1, \ldots, N\}$.

The above assumption is easy to satisfy. Specifically, Gaussian noise and uniform noise with non-negative expectation are examples, which are two popular distributions applied in neural networks [10, 8].

3.2. Lower Uncertainty

Under the above two assumptions, we state our main result, which shows that the uncertainty that the proposed multiplicative noise introduces into the system is less than the additive noise. In other words, under the Assumption 1 and 2, the channel capability of $\mathbf{X}_i + \mathbf{Z}_i$ is higher than $\mathbf{X}_i + \mathbf{Z}_i^{'}$. Here, $\mathbf{Z}_i$ is the additive noise and $\mathbf{Z}_i^{'}$ is the multiplicative noise. $P(\mathbf{Z}_i^{'}) = P(\mathbf{Z}_i)P(\mathbf{X}_i)$, and $\mathbf{X}_i$ and $\mathbf{Z}_i$ are independent for all $i = \{1, \ldots, N\}$.

3.2.1 Uncertainty of Additive Noise Channel

Given $\mathbf{X}_i$ and $\mathbf{Z}_i$ whose probability distribution satisfies Assumption 1 and 2. First, the channel capability of $\mathbf{X}_i + \mathbf{Z}_i$ is given by

$$\begin{aligned} C_{i1} &= \max_{p_i(x) \text{ s.t. } \mathbb{E}(\mathbf{X}_i^2) \leq \text{Pw}} I(\mathbf{X}_i + \mathbf{Z}_i; \mathbf{Z}_i) \\ &= \max_{p_i(x) \text{ s.t. } \mathbb{E}(\mathbf{X}_i^2) \leq \text{Pw}} h(\mathbf{X}_i + \mathbf{Z}_i) - h(\mathbf{X}_i + \mathbf{Z}_i | \mathbf{X}_i) \end{aligned}$$

where h represents the Shannon entropy. Since $\mathbf{Z}_i$ is independent with $\mathbf{X}_i$, we have

$$C_{i1} = \max_{p_i(x) \text{ s.t. } \mathbb{E}(\mathbf{X}_i^2) \leq \text{Pw}} h(\mathbf{X}_i + \mathbf{Z}_i) - h(\mathbf{Z}_i).$$

Note that $\mathbb{E}((\mathbf{X}_i + \mathbf{Z}_i)^2)$=$\mathbb{E}(\mathbf{X}_i^2) + 2\mathbb{E}(\mathbf{X}_i)E(\mathbf{Z}_i) + \mathbb{E}(\mathbf{Z}_i^2)$. Under Assumption 2, $\mathbb{E}((\mathbf{X}_i + \mathbf{Z}_i)^2)$ has a upper bound of $B(Pw, \mathbf{Z}_i) \triangleq Pw + 2\sqrt{Pw}E(\mathbf{Z}_i) + \mathbb{E}(\mathbf{Z}_i^2)$ given $\mathbf{Z}_i$. Here, the bound is attained when $\mathbb{E}(\mathbf{X}_i) = \sqrt{Pw}$.

Recall the property of differential entropy [23]

$$h(\mathbf{X}_i) \leq \frac{1}{2} \log 2\pi e K$$

where the equality is attained if and only if $\mathbf{X}_i$ is Gaussian distributed and K is the co-variance matrix. C_{i1} can be expressed as

$$C_{i1} = \frac{1}{2} \log 2\pi e B(Pw, \mathbf{Z}_i) - h(\mathbf{Z}_i)$$

where $\mathbf{Z}_i$ is a given noise.

3.2.2 Uncertainty of Multiplicative Noise Channel

We further compute the channel capability of $\mathbf{X}_i + \mathbf{Z}_i^{'}$, where $\mathbf{Z}_i^{'} = \mathbf{X}_i \mathbf{Z}_i$.

$$\begin{aligned} C_{i2} &= \max_{p_i(x) \text{ s.t. } \mathbb{E}(\mathbf{X}_i^2) \leq \text{Pw}} I(\mathbf{X}_i + \mathbf{Z}_i^{'}; \mathrm{Z}_i^{'}) \\ &= \max_{p_i(x) \text{ s.t. } \mathbb{E}(\mathbf{X}_i^2) \leq \text{Pw}} h(\mathbf{X}_i + \mathbf{Z}^{'}) - h(\mathbf{X} + \mathbf{Z}^{'} | \mathbf{X}) \\ &= \max_{p_i(x) \text{ s.t. } \mathbb{E}(\mathbf{X}_i^2) \leq \text{Pw}} h(\mathbf{X}_i + \mathbf{X}_i \mathbf{Z}_i) - h(\mathbf{X}_i \mathbf{Z}_i | \mathbf{X}) \\ &= \max_{p_i(x) \text{ s.t. } \mathbb{E}(\mathbf{X}_i^2) \leq \text{Pw}} h(\mathbf{X}_i + \mathbf{X}_i \mathbf{Z}_i) - h(\mathbf{Z}_i) \end{aligned}$$

Note that $\mathbb{E}((\mathbf{X}_i + \mathbf{Z}_i^{'})^2)$=$\mathbb{E}(\mathbf{X}_i^2) + \mathbb{E}(\mathbf{X}_i^2)\mathbb{E}(\mathbf{Z}_i^2) + 2\mathbb{E}(\mathbf{X}_i^2)\mathbb{E}(\mathbf{Z}_i)$. Under Assumption 2, $\mathbb{E}((\mathbf{X}_i + \mathbf{Z}_i^{'})^2)$ has a upper bound of $B(Pw, \mathbf{Z}_i^{'}) \triangleq Pw + 2Pw\mathbb{E}(\mathbf{Z}_i) + Pw\mathbb{E}(\mathbf{Z}_i^2)$ given $\mathbf{Z}_i$. Here, the bound is attained when $\mathbb{E}(\mathbf{X}_i) = \sqrt{Pw}$.

According to the property of differential entropy, C_{i2} can be further written as

$$C_{i2} = \frac{1}{2} \log 2\pi e B(Pw, \mathbf{Z}_i^{'}) - h(\mathbf{Z}_i).$$

3.2.3 Analysis of Lower Uncertainty

Finally, we can show the channel capability of $\mathbf{X}_i + \mathbf{Z}_i^{'}$ is lower than the channel capability of $\mathbf{X}_i + \mathbf{Z}_i$ since

$$\begin{aligned} C_{i2} - C_{i1} &= \frac{1}{2} \log 2\pi e \left(B(Pw, \mathbf{Z}_i^{'}) - B(Pw, \mathbf{Z}_i) \right) \\ &= 2\mathbb{E}(\mathbf{Z}_i)(Pw - \sqrt{Pw}) + \mathbb{E}(\mathbf{Z}_i^2)(Pw - 1) \\ &\leq 0. \end{aligned}$$

Under Assumption 1 and 2, the equality of the above formulation holds when $Pw = 1$ or $\mathbb{E}(\mathbf{X}_i) = \mathbb{E}(\mathbf{X}_i^2) = 0$. Since each element of $\mathbf{X}$ is independent, the output of the multiplicative noisy vector is of the form $\mathbf{Y} = \mathbf{X} + \mathbf{X} \odot \mathbf{Z}$ where each element of $\mathbf{Z}$ is i.i.d. distributed. Here $\odot$ represents element-wise multiplication. In the following section, we investigate one element of the vector $\mathbf{X} + \mathbf{X} \odot \mathbf{Z}$, i.e., $x + xz$.

4. Lower Effects of Mixing

Let $\mathbf{X}^i$ and $\mathbf{X}^j$ be two different training data $i \neq j$. We denote $\mathbf{Z}^i$ and $\mathbf{Z}^j$ as the latent vectors of $\mathbf{X}^i$ and $\mathbf{X}^j$, respectively. Both $\mathbf{Z}^i$ and $\mathbf{Z}^j$ are high dimensional vectors. Without loss of generality, let x_i and x_j be any element of $\mathbf{Z}^i$ and $\mathbf{Z}^j$. A positive value d is defined as following. Generally, the inserted noise mixes x_i and x_j as $x_i + z_i = x_j + z_j$, where $|x_i - x_j| \leq d$, and z_i and z_j are the noise terms sampled from the same probabilistic distribution $p(z)$. Moreover, x_i and x_j can not be mixed when $| x_i - x_j | > d$. Here, the value of d is inversely proportional to the goodness of the inserted noise. Obviously, Gaussian noise $\mathcal{N}(\mu, \sigma)$ is not a good choice since it dose not have a bounded support. However, the noise $\mathcal{B}(-a, b)$ can be used where the range of noise is $[-a, b]$, $a > 0$, $b > 0$. An analysis is given that $x + xz$ has a smaller d than $x + z$ when z satisfies a range assumption shown below.

A practical assumption is proposed for z, and it imposes restrictions on the noise that is inserted into the deep neural network.

Assumption 3 *z has a support in $[-a, b]$, where $1 > a > 0$ and $1 > b > 0$.*

4.1. Analysis of Mixing

Under Assumption 1 to 3, we can show that $x + xz$ has a smaller expectation of d than $x + z$. In other words, it gives the fact that $x + xz$ has lower effects of mixing than $x + z$. That is, $x + xz$ has a smaller $\mathbb{E}(d)$ than that of $x + z$.

4.1.1 Effects of Additive Noise

Without loss of generality, let $\mathbb{E}(x_i) < \mathbb{E}(x_j)$. The expectation of d for additive noisy input $x + z$ is computed as following. Under Assumption 3, $\mathbb{E}(x_i + z) \in [\mathbb{E}(x_i) - a, \mathbb{E}(x_i) + b]$, and $\mathbb{E}(x_j + z) \in [\mathbb{E}(x_j) - a, \mathbb{E}(x_j) + b]$. Let $\mathbb{E}(x_i) + b = \mathbb{E}(x_j) - a$. We have $\mathbb{E}(d_1) = | \mathbb{E}(x_j) - \mathbb{E}(x_i) | = a + b$.

4.1.2 Effects of Proposed Multiplicative Noise

The value of d for $x + xz$ is computed as following. First, under Assumption 1, let $0 \leq \mathbb{E}(x_i) < \mathbb{E}(x_j) \leq 1$, then $\mathbb{E}(x_i + x_i z) \in [\mathbb{E}(x_i) - a\mathbb{E}(x_i), \mathbb{E}(x_i) + b\mathbb{E}(x_i)]$, and $\mathbb{E}(x_j + x_j z) \in [\mathbb{E}(x_j) - aE(x_j), \mathbb{E}(x_j) + b\mathbb{E}(x_j)]$. Let $\mathbb{E}(x_j) - a\mathbb{E}(x_j) = \mathbb{E}(x_i) + b\mathbb{E}(x_i)$. We have $\mathbb{E}(d_2) = | \mathbb{E}(x_j) - \mathbb{E}(x_i) | = a\mathbb{E}(x_i) + b\mathbb{E}(x_j)$

Second, under Assumption 1, let $-1 < \mathbb{E}(x_i) < 0 < \mathbb{E}(x_j) < 1$, $\mathbb{E}(x_i + x_i z)$ and $\mathbb{E}(x_j + x_j z)$ are in $[\mathbb{E}(x_i) + bE(x_i), \mathbb{E}(x_i) - a\mathbb{E}(x_i)]$ and $[\mathbb{E}(x_j) - a\mathbb{E}(x_j), \mathbb{E}(x_j) + b\mathbb{E}(x_j)]$, respectively. Under Assumption 3, $a \leq 1$, for all $d \leq 0$, we have $\mathbb{E}(x_j) - a\mathbb{E}(x_j) \leq \mathbb{E}(x_i) - a\mathbb{E}(x_j)$. Therefore, $\mathbb{E}(d_3) = 0$.

Third, under Assumption 1, let $-1 \leq \mathbb{E}(x_i) < \mathbb{E}(x_j) \leq 0$, $\mathbb{E}(x_i + x_i z)$ and $\mathbb{E}(x_j + x_j z)$ are in $[\mathbb{E}(x_i) + b\mathbb{E}(x_i), \mathbb{E}(x_i) - a\mathbb{E}(x_i)]$ and $[\mathbb{E}(x_j) + b\mathbb{E}(x_j), \mathbb{E}(x_j) - a\mathbb{E}(x_j)]$, respectively. We thus have $\mathbb{E}(d_4) = | \mathbb{E}(x_i) - \mathbb{E}(x_j) | = -a\mathbb{E}(x_i) - b\mathbb{E}(x_j)$.

4.1.3 Lighter Effects of Mixing

To summarize, under Assumption 1 to 3, we have

$$\begin{cases} \text{if } 0 \leq \mathbb{E}(x_i) < \mathbb{E}(x_j) \leq 1, & \text{then } \mathbb{E}(d_1) > \mathbb{E}(d_2); \\ \text{if } -1 \leq \mathbb{E}(x_i) < 0 \leq \mathbb{E}(x_j) \leq 1, & \text{then } \mathbb{E}(d_1) > \mathbb{E}(d_3); \\ \text{if } -1 \leq \mathbb{E}(x_i) < \mathbb{E}(x_j) \leq 0, & \text{then } \mathbb{E}(d_1) > \mathbb{E}(d_4). \end{cases}$$

Similarly, a similar conclusion can be made when $\mathbb{E}(x_i) > \mathbb{E}(x_j)$.

Following the above analysis, one implementation is $\mathbf{X} + \alpha\mathbf{X} \odot \mathbf{Z}$, where $\mathbf{Z}$ is a multi-dimensional vector with each element of $\mathbf{Z}$ i.i.d distributed. The multiplicative noise term is $\alpha\mathbf{X} \odot \mathbf{Z}$. In the following experiments, the hype-parameter α is set to 0.01, and the probability distribution for each element of $\mathbf{Z}$ is $\mathcal{U}(-1, 1)$.

5. Experiments

In this section, two experiments are conducted. In the first experiment, the multiplicative noise term is inserted in the GAN architectures under Assumption 1 and 2. In particular, the proposed noise term is inserted before the output layer of the discriminator and before the output layer of the generator. Normally, the value of pixels of output images are in the range of $[0, 1]$ and the value of the discriminator is in the range of $[0, 1]$. Assumption 1 and 2 are satisfied at these two positions. In the second experiment, the proposed noise term is inserted after the input layer of the discriminator (temporal network [11]) and before the output layer of the generator.

5.1. Two-level Unsupervised Generation from Two-level Latent Space

Current GAN architectures are able to generate real visual images from a latent probabilistic space. This unsupervised generation is formulated as $\mathbb{Y} = G(\mathbb{Z})$, where $\mathbb{Y}$ represents a set of real images and $\mathbb{Z}$ represents a set of multi-dimensional samples from a latent probabilistic

space. However, to our best knowledge, a simple hierarchical generation of real images is not presented through GANs. In this experiment, it is shown that the application of simple multiplicative noise into GANs enables GANs to generate a two-lever unsupervised generation to some extend. Furthermore, it is shown that the visual quality and visual variety of the generated samples are improved compared with two base models, including GAN+AN (additive Noise) and WGAN+AN (additive Noise).

5.1.1 Definition of The Two-lever Unsupervised Generation

The two-lever unsupervised generation in this experiment can be expressed as following.

$$\begin{cases} \mathbb{Y} = G(\mathbb{Z}); \\ \mathbb{Y}_1 = G(\mathbb{Z}_1); \\ \mathbb{Y}_2 = G(\mathbb{Z}_2), \end{cases}$$

where $\mathbb{Z}_1 \cap \mathbb{Z}_2 = \varnothing$, $\mathbb{Z}_1 \subset \mathbb{Z}$, $\mathbb{Z}_2 \subset \mathbb{Z}$, $\mathbb{Y}_1 \cap \mathbb{Y}_2 = \varnothing$, $\mathbb{Y}_1 \subset \mathbb{Y}$ and $\mathbb{Y}_2 \subset \mathbb{Y}$. $\mathbb{Z}_1$ and $\mathbb{Z}_2$ are two sub-trees of $\mathbb{Z}$, and $\mathbb{Y}_1$ and $\mathbb{Y}_2$ are two sub-trees of $\mathbb{Y}$.

5.1.2 Unsupervised Generation of Human Faces

In this experiment, CelebFaces Attribute Dataset (CelebA) is used for unsupervised generation of human faces. As presented in unsupervised generation via GAN+AN [8] and WGAN+AN [1], $\mathbb{Z}$ contains the latent vector $\mathbf{Z}$ which is a 100-dimensional random vector drawn from $\mathcal{U}(-1, 1)$, and $\mathbb{Y}$ contains the generated images $\mathbf{Y}$ which is a real human face. The multiplicative noise term is added after the output layer of the discriminator of each GAN architecture, and the new architectures are named as GAN+dc and WGAN+dc. Similarly, the new GAN architecture is named as GAN+gc and WGAN+gc where the multiplicative noise channel is used before the output layer of the generator. A generation structure is obtained via the same unsupervised training process with GAN+AN [8] and WGAN+AN [1], respectively. The batch size, epochs of training and learning rate remain the same.

Figure 1, Figure 2 and Figure 3 represent unsupervised human faces generation through GAN+AN, GAN+dc and GAN+gc, respectively. As shown, the top-lever generated human faces through GAN+dc and GAN+gc obtain higher visual quality and variety than GAN+AN. For example, some of the generated faces at the top-lever are wrapped in Figure 1. However, GAN+dc and GAN+gc generate top-lever faces without distortion. Moreover, the visual variety is also improved since GAN+dc and GAN+gc are capable of generating faces with hats, glasses, different face outlines and different poses more likely, while GAN+AN generates human faces with similar face outlines. Furthermore, only

Figure 1. Human face generation. The architecture is GAN+AGN. The left sub-tree $\mathbf{Y}_1 = G(\mathbf{Z}_1)$ and the right sub-tree $\mathbf{Y}_2 = G(\mathbf{Z}_2)$ are a mess. The central tree $\mathbf{Y} = G(\mathbf{Z})$ shows the generated human face, and some of the generated faces are wrapped.

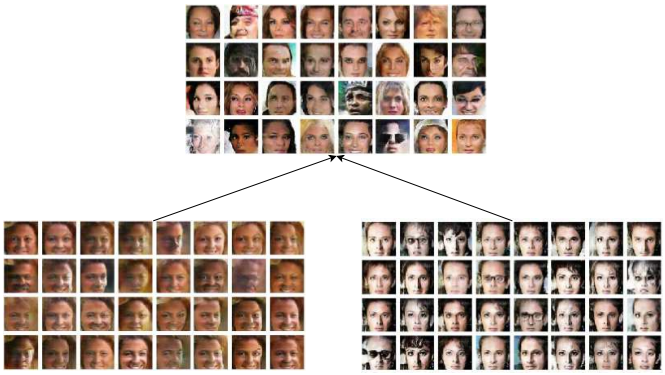

Figure 2. Two-level hierarchical human face generation via gender. The architecture is GAN+dc. The left sub-tree $\mathbf{Y}_1 = G(\mathbf{Z}_1)$ and the right sub-tree $\mathbf{Y}_2 = G(\mathbf{Z}_2)$ show the generated human faces of different face outlines respectively. The central tree $\mathbf{Y} = G(\mathbf{Z})$ shows the generated human faces are of much more variety and have much higher visual quality.

fixed human faces features are generated through GAN+AN at the bottom lever, while human faces of a sizable variety are generated at the bottom lever.

Similarly, in the comparison among Figure 4, Figure 5 and Figure 6, WGAN+dc and WGAN+gc also improve the visual quality and visual variety of generated faces than WGAN+AN. Furthermore, the generated faces at the bottom lever are shown to obtain discrimination of gender through WGAN+dc and WGAN+gc. In particular, as Figure 6 shows, the left bottom represents generated female faces and the right bottom represents generated male faces. While in Figure 4, features of male faces are generated at the bottom lever. It is worth noted that the experiment setup for GAN+dc and GAN+gc is the same as the one in GAN [8], and the experiment setup for WGAN+dc and WGAN+gc is the same as the one in WGAN [1].

Figure 3. Two-level hierarchical human face generation via gender. The architecture is GAN+gc. The left sub-tree $\mathbf{Y}_1 = G(\mathbf{Z}_1)$ and the right sub-tree $\mathbf{Y}_2 = G(\mathbf{Z}_2)$ show the generated human faces of different genders respectively. The central tree $\mathbf{Y} = G(\mathbf{Z})$ shows the generated human faces of much more variety and much higher visual quality.

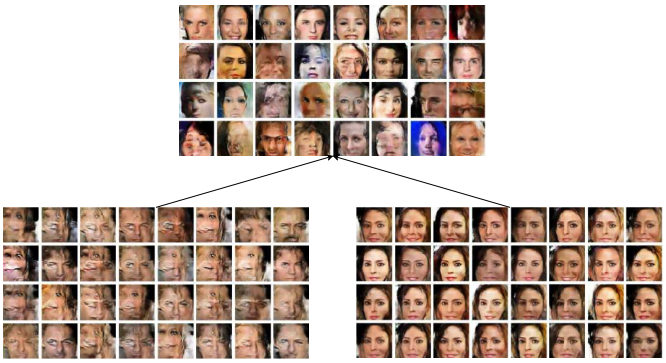

Figure 4. Human face generation. The center architecture is the generation via WGAN+AGN. The left sub-tree $\mathbf{Y}_1 = G(\mathbf{Z}_1)$ shows the generated high-lever features of male human faces. The right sub-tree $\mathbf{Y}_2 = G(\mathbf{Z}_2)$ shows the generated female faces. The central tree $\mathbf{Y} = G(\mathbf{Z})$ shows the generated human faces with a much more variety.

Figure 5. Two-level hierarchical human face generation via gender. The architecture is the generation via WGAN+dc. The left sub-tree $\mathbf{Y}_1 = G(\mathbf{Z}_1)$ shows the generated female faces. The right sub-tree $\mathbf{Y}_2 = G(\mathbf{Z}_2)$ shows the generated male faces (some faces are wrapped). The central tree $\mathbf{Y} = G(\mathbf{Z})$ shows the generated human faces with a much more variety.

Figure 6. Two-level hierarchical human face generation via gender. The center architecture is the generation via WGAN+gc. The left sub-tree $\mathbf{Y}_1 = G(\mathbf{Z}_1)$ shows generated male faces. The right sub-tree $\mathbf{Y}_2 = G(\mathbf{Z}_2)$ shows the generated female faces. The center $\mathbf{Y} = G(\mathbf{Z})$ tree shows the generated human faces with a much more variety and much higher visual quality.

5.2. Semi-supervised Classification

The GANs architectures can be applied to semi-supervised classification of images. In this experiment, the baseline1 is the Improve-GAN architecture with additive noise [20]. The multiplicative noise term is added after the input layer of the discriminator and before the output layer of the generator. These architectures are named as baseline1+dc and baseline1+gc, respectively. Besides, the proposed noise term is added after the input layer of the π-model with additive noise (baseline2) [11], and it is named as baseline2+dc. Moreover, the proposed noise term is added after the input layer of the temporal model with additive noise (baseline3) [11], and it is named as baseline3+dc. The data augmentation is termed as Au if the same standard augmentation is used in [11] including ZCA whiten, random crops and flips. Experiments on MNIST, SVHN and CIFAR-10 are conducted.

5.2.1 Results on CIFAR-10, SVHN and MNIST

The CIFAR-10 dataset contains $50,000$ training images and $10,000$ test images. These images are on ten image categories. The SVHN dataset is the street view house number dataset, and this dataset contains 32×32 color images of numbers. The training set has $73,257$ house numbers while the test set has $26,032$ house numbers. The MNIST dataset contains $60,000$ images of digits. There are $50,000$ images of digits in the training set and $10,000$ images of digits in the testing set. The images are binary images, and the size of each image is 28×28.

In the semi-supervised classification experiments, labels of a small number of images are used. Experiments are conducted with 400 labeled examples per category for CIFAR-

Table 1. Semi-supervised classification results on MNIST

Methods	MNIST Error (10 per class)
CatGAN	$1.39\% \pm 0.28\%$
ALI	–
IGAN (Baseline1)	$0.93\% \pm 0.06\%$
π-model (Baseline2)	–
π-model (Baseline2)[Au]	–
Temporal-model (Baseline3)[Au]	–
Baseline1+gc	$0.99\% \pm 0.02\%$
Baseline1+dc	$\mathbf{0.88}\% \pm \mathbf{0.02}\%$
Baseline2+dc	–
Baseline2+dc[Au]	–
Baseline3+dc[Au]	–

Table 2. Semi-supervised classification results SVHN

Methods	SVHN Error (100 per class)
CatGAN	–
ALI	7.3%
IGAN (Baseline1)	$8.11\% \pm 1.3\%$
π-model (Baseline2)	$5.43\% \pm 0.25\%$
π-model (Baseline2)[Au]	$4.82\% \pm 0.17\%$
Temporal-model (Baseline3)[Au]	$4.42\% \pm 0.16\%$
Baseline1+gc	$\mathbf{6.35}\% \pm \mathbf{0.02}\%$
Baseline1+dc	$\mathbf{5.95}\% \pm \mathbf{0.02}\%$
Baseline2+dc	$\mathbf{5.10}\% \pm \mathbf{0.12}\%$
Baseline2+dc[Au]	$\mathbf{4.60}\% \pm \mathbf{0.09}\%$
Baseline3+dc[Au]	$\mathbf{4.19}\% \pm \mathbf{0.06}\%$

10, 100 labeled examples per category for SVHN and 10 labeled examples per category for MNIST. In the experiments for comparison, all of the proposed models are trained in the same experiment configuration with baseline1 [20], baseline2 [11] and baseline3 [11], respectively. The parameters for the training are also the same as the baselines.

Table 1 compares the results on MNIST. Compared with baseline1 (Improve-GAN), semi-classification accuracy improvement is achieved through using the simple multiplicative noise. Particularly, it's improved from 0.99% to $\mathbf{0.88}\%$. The experiments are run 10 times, and the average value and the standard deviation are calculated.

Table 2 compares the results on SVHN. Compared with the baseline1 (Improve-GAN), baseline2 (π-model) and baseline3 (Temporal-model), accuracy improvement is achieved through using the simple multiplicative noise term. Particularly, it is improved from 8.11% to $\mathbf{6.35}\%$ and $\mathbf{5.95}\%$ for Baseline1, from 5.43% to $\mathbf{5.10}\%$ for Baseline2, from 4.82% to $\mathbf{4.60}\%$ for Baseline2 [Au], from 4.42% to $\mathbf{4.19}\%$ for Baseline3 [Au]. All the experiments are run 10 times, and the average value and the standard deviation are calculated.

Table 3. Semi-supervised classification results CIFAR-10

Methods	CIFAR-10 Error (400 per class)
CatGAN	$19.60\% \pm 0.40\%$
ALI	18.30%
IGAN (Baseline1)	$18.60\% \pm 2.30\%$
π-model (Baseline2)	$16.55\% \pm 0.29\%$
π-model (Baseline2)[Au]	$12.36\% \pm 0.31\%$
Temporal-model (Baseline3)[Au]	$12.16\% \pm 0.24\%$
Baseline1+gc	$\mathbf{16.95}\% \pm \mathbf{0.05}\%$
Baseline1+dc	$\mathbf{16.75}\% \pm \mathbf{0.06}\%$
Baseline2+dc	$\mathbf{16.24}\% \pm \mathbf{0.09}\%$
Baseline2+dc[Au]	$\mathbf{12.05}\% \pm \mathbf{0.07}\%$
Baseline3+dc[Au]	$\mathbf{11.91}\% \pm \mathbf{0.04}\%$

Table 3 compares the results on CIFAR-10. Compared with the baseline1 (Improve-GAN), baseline2 (π-model) and baseline3 (Temporal-model), accuracy improvement is achieved through using the simple multiplicative noise term. Particularly, it is improved from 18.60% to $\mathbf{16.95}\%$ and $\mathbf{16.75}\%$ for Baseline1, from 16.55% to $\mathbf{16.24}\%$ for Baseline2, from 12.36% to $\mathbf{12.05}\%$ for Baseline2 [Au], from 12.16% to $\mathbf{11.91}\%$ for Baseline3 [Au]. All the experiments are run 10 times, and the average value and the standard deviation are calculated.

In addition, new records for semi-classification accuracy on CIFAR (4000), SVHN (1000) and MNIST (100) are achieved. They are $\mathbf{11.91}\%$, $\mathbf{4.19}\%$ and $\mathbf{0.88}\%$, respectively.

6. Discussion

In this paper, multiplicative noise is introduced and applied to generative adversarial networks (GANs). It is demonstrated to obtain two advantages including lower introduction of uncertainty and less effects of mixing in comparison with the additive noise under mild assumptions applicable for GANs. Moreover, a simple term of multiplicative noise is applied without extra regularization for unsupervised face generation and semi-classification. Extensive experiments are conducted to show that the visual quality and variety, and the semi-classification accuracy are improved in the comparison with the additive noise.

This work an initial attempt to explore the role of noise in deep neural networks. The proposed formulation of the multiplicative noise is simple. We apply this simple noise to show that multiplicative noise obtain several advantages than the additive noise in the framework of GAN. It is shown that different forms of noise rather than the additive noise may be beneficial to the performance of deep neural network. However, the proposed method works under mild assumptions, and it is only proposed for unsupervised deep neural networks. In the near future, different and effective formulations of multiplicative noise may be used in a larger

range of deep neural networks and relative tasks.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] K. Audhkhasi, O. Osoba, and B. Kosko. Noise benefits in backpropagation and deep bidirectional pre-training. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8. IEEE, 2013.

[3] K. Audhkhasi, O. Osoba, and B. Kosko. Noise-enhanced convolutional neural networks. *Neural Networks*, 78:15–23, 2016.

[4] B. B. Averbeck, P. E. Latham, and A. Pouget. Neural correlations, population coding and computation. *Nature reviews neuroscience*, 7(5):358–366, 2006.

[5] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1613–1622, 2015.

[6] I. Durugkar, I. Gemp, and S. Mahadevan. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673*, 2016.

[7] A. Ghosh, V. Kulharia, and V. Namboodiri. Message passing multi-agent gans. *arXiv preprint arXiv:1612.01294*, 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 *Advances in neural information processing systems*, pages 2672–2680, 2014.

[9] G. E. Hinton. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986.

[10] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[11] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[12] C. Li, K. Xu, J. Zhu, and B. Zhang. Triple generative adversarial nets. *arXiv preprint arXiv:1703.02291*, 2017.

[13] J. Luo. Learning inverse mapping by autoencoder based generative adversarial nets. *arXiv preprint arXiv:1703.10094*, 2017.

[14] X. Mao, Q. Li, and H. Xie. Aligngan: Learning to align cross-domain images with conditional generative adversarial networks. *arXiv preprint arXiv:1707.01400*, 2017.

[15] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. *arXiv preprint ArXiv:1611.04076*, 2016.

[16] S. Pascual, A. Bonafonte, and J. Serrà. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

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

[18] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From source to target and back: symmetric bi-directional adaptive gan. *arXiv preprint arXiv:1705.08824*, 2017.

[19] 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 2234–2242, 2016.

[20] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016.

[21] M. N. Shadlen and W. T. Newsome. Noise, neural codes and cortical organization. *Current opinion in neurobiology*, 4(4):569–579, 1994.

[22] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. *arXiv preprint arXiv:1612.07828*, 2016.

[23] J. V. Stone. *Information theory: a tutorial introduction*. Sebtel Press, 2015.

[24] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[25] C. Wang and J. C. Principe. Training neural networks with additive noise in the desired signal. *IEEE Transactions on Neural Networks*, 10(6):1511–1517, 1999.

[26] R. Wang, A. Cully, H. J. Chang, and Y. Demiris. Magan: Margin adaptation for generative adversarial networks. *arXiv preprint arXiv:1704.03817*, 2017.

[27] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

[28] H. Yin, F. S. Melo, A. Billard, and A. Paiva. Associate latent encodings in learning from demonstrations. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, number EPFL-CONF-224072, 2017.

[29] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.

[30] R. M. Zur, Y. Jiang, L. L. Pesce, and K. Drukker. Noise injection for training artificial neural networks: A comparison with weight decay and early stopping. *Medical physics*, 36(10):4810–4818, 2009.