JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Deploying diffusion based generation model for improved training of GAN

[1] Yoga Harshitha Duddukuri
[1] Student, M.Tech
[1]Department of computer science
[1]GITAM Deemed University, Hyderabad, India

Abstract : This paper discovers the usage and studies the effect of diffusion based generation model to transform the raw training data of GAN into standard normal latent space. This improved the training stability and reduced resource utilization and played a crucial role in decreasing the training error. Papers till date have explored the use of generating models for generating the data it is trained on. The research done on natural language processing shows the significance of word embedding and transforming unstructured data into trainable domain [1] , which involve in lot of resources for training, and there are very few papers on generalizing the process, as compared to the papers that exists on its applications. In this paper we are using diffusion based generation model as intermediate processing block on USDA food database. This offers insights into extending the application of GAN networks into fields that did not gain much attention till date. The main application is to prepare a meal plan by generating various combinations of dishes in the database so as to meet the nutritional needs. GAN that is trained directly on the food database was unstable and generated a huge error. The standard normalizing and de normalizing techniques were barely up to the help in decreasing the error. Instead of training the GAN to directly generate the data, we have stacked already trained diffusion based generation model at the end of GAN, that was trained to map the food data into latent space, and trained GAN network for time series generation of different food combinations that would meet the nutritional requirements.

IndexTerms* - *diffusion based generation, improving the training of GAN, GAN applications, normalization, embedding vectors, data processing.

I. Introduction

The aim is to generate a meal plan that meet the nutritional requirements. Food database used is the USDA nutritional data for SR legacy food. The objective is to train the neural network to generate meal plan, food selected from data with portion assigned, based on this data, given the limits in which the meal generated should obey. Several machine learning architectures were considered before finalizing the architecture [2] . GAN is deployed for generating meal.

A. Literature survey

generative modelling has numerous direct applications including image synthesis: super-resolution, text-to-image and image-to-image conversion, inpainting, attribute manipulation, pose estimation; video: synthesis and retargeting; audio: speech and music synthesis; text: summarisation and translation; reinforcement learning; computer graphics: rendering, texture generation, character movement, liquid simulation; medical: drug synthesis, modality conversion; and out-of-distribution detection. The central idea of generative modelling stems around training a generative model whose samples $x\sim \sim p\theta(x\sim)$ come from the same distribution as the training data distribution, $x \sim pd(x)$ [2] .

II. Network Architecture

A. Block diagram

The architecture is carefully designed to meet the objective with minimum resources.

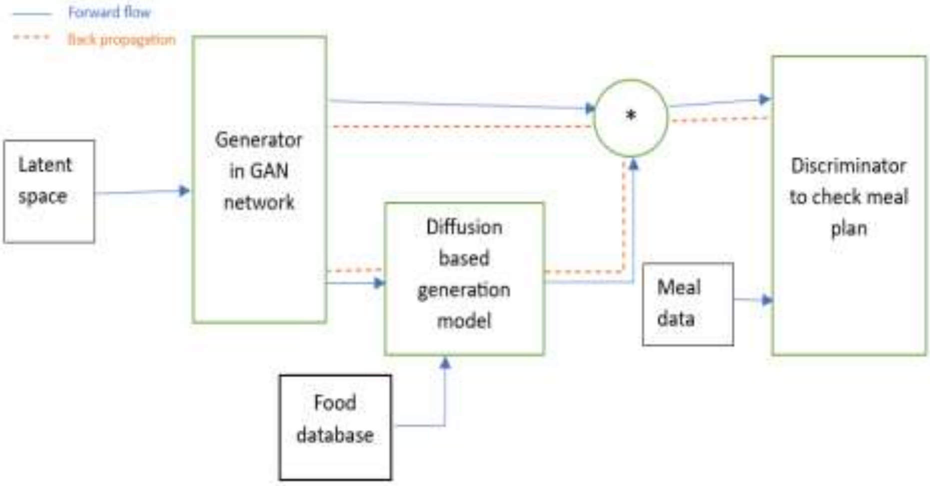

Fig 2.1 Network architecture

B. *Flow of the process*

In a nut shell, the main objective of the network is to generate list of food items that meet the nutrient criteria. GAN is employed to get the job done. The uneven nature of the data and its sparsity makes it difficult for training. Data is transformed into another latent space using diffusion based generation model, which is prep trained on the desired data (here it is USDA SR legacy food). GAN is used to predict the food along its weight, which is then fed to the discriminator.

III. MATHEMATICAL INSIGHT

Diving into the mathematical equations that govern the network, detailed mathematics of simple WGAN that is used for generating the data is given in the following sections. The discriminator is trained straight forward, with the nutritional limits, so as to avoid overfitting of the data. It is iteratively trained to improve accuracy. Hence smoothing the training process. Generalizing the discriminator speed up the process and reduces possibility of training instability.

A. *GAN and its training*

GANs consist of two networks, a discriminator D:Rn→[0,1] which estimates the probability that a sample comes from the data distribution x ∼ pd(x), and a generator G:Rm→Rn which given a latent variable z ∼ pz(z), captures pd by tricking the discriminator into thinking its samples are real. D is trained to distinguish between real and fake, samples generated by G, while G is trained to minimize the distance between real and its generated samples. This can be interpreted as D and G playing a mini-max game, as with prior work [192], [193], optimizing the value function V(G,D)

$$\min_G \max_D V(G,D) = \mathbb{E}_{\boldsymbol{x}\sim p_d(\boldsymbol{x})}[\ln D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\ln(1-D(G(\boldsymbol{z})))].$$

[3] (1)

B. *How is the GAN trained*

Adversarial nature of GANs makes them notoriously difficult to train; Nash equilibrium is hard to achieve, since non-cooperation cannot guarantee convergence, thus training often results in oscillations of increasing amplitude. [2] . Through the analysis of the loss function, it is concluded that the training of the module is guided only by the distributed distance loss function. For the initial training discriminator, it does not learn enough characteristic boundaries of real data, and there is no essential description of the data. [4].

Here we have enforced 1-lipschitz condition on to the GAN [5]. Instead of considering entropy loss, we compute Wasserstein distance and gradients are computed with respect to this loss function

$$W(\mathbb{P}_r,\mathbb{P}_g) = \inf_{\gamma\in\Pi(\mathbb{P}_r,\mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\,\|x-y\|\,\big]\ ,$$

(2)

where Π(Pr, Pg) denotes the set of all joint distributions γ(x, y) whose marginals are respectively Pr and Pg. Intuitively, γ(x, y) indicates how much “mass” must be transported from x to y in order to transform the distributions Pr into the distribution Pg. The EM distance then is the “cost” of the optimal transport plan.[9]

C. *Diffusion based generation*

Data to be generated is sparse and requires to be standardized. Diffusion based generation model uses a stochastic differential equation (SDE) that smoothly transforms a complex data distribution to a known prior distribution by slowly injecting noise, and a corresponding reverse-time SDE that transforms the prior distribution back into the data distribution by slowly removing the noise.[8].

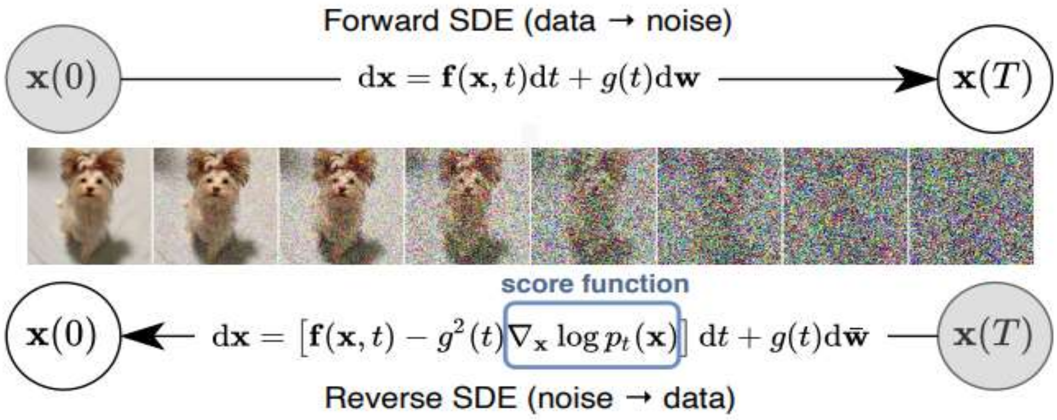

Fig 3.1 SDE of diffusion based model [8]

D. Putting it all together

In this case the function of discriminator is made quite straight forward. It can be achieved by second order function, whose parameters are fine tuned to provide correct feedback to the generator. Discriminator is trained initially on the all possible data and first layers are freeze, then is fine tuned, with generator outputs, along with the generator.

$$\max_D V(G, D) = \mathbb{E}_{\boldsymbol{x}\sim p_d}[\ln D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\ln(1 - D(\boldsymbol{x}))] \quad (3)$$

target data distribution is $y \sim N(\text{td},1)$

Generator is trained to minimise the distance. The most common phenomenon of dying gradients is taken care by standardizing the data

Target of the generator is the $x(T) \sim p_T$, of the diffusion model. Reverse diffusion is used to extract the data and is fed to the discriminator.

Gradients computed by the discriminator are fed back to the generator for training.

E. Data analysis

Database used here is USDA SR legacy food.

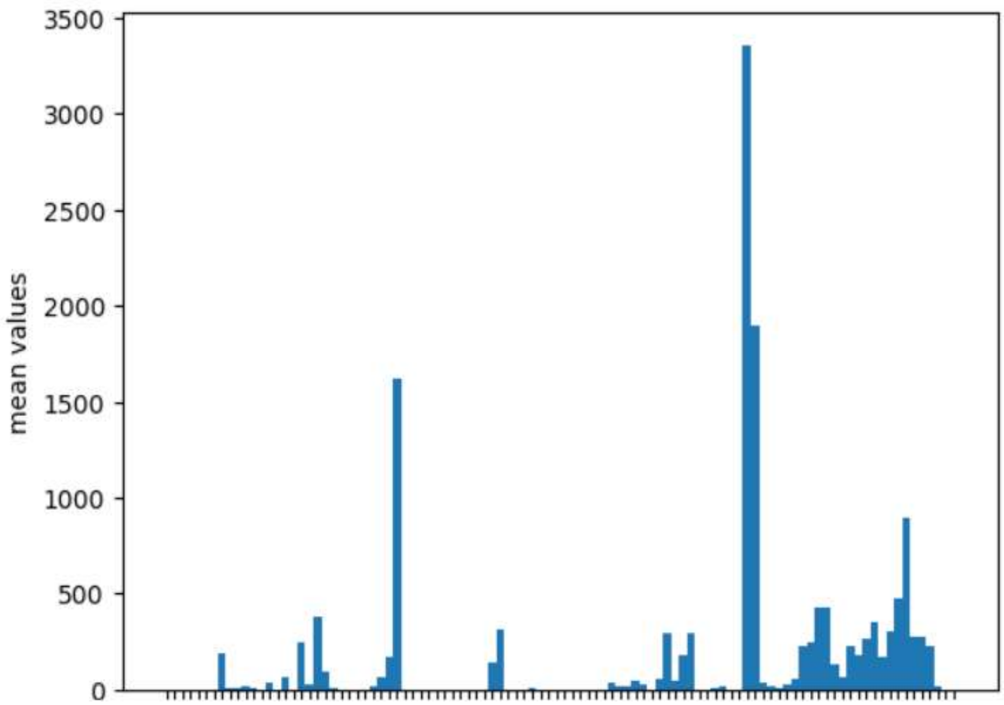

Fig 3.1 mean distribution

–

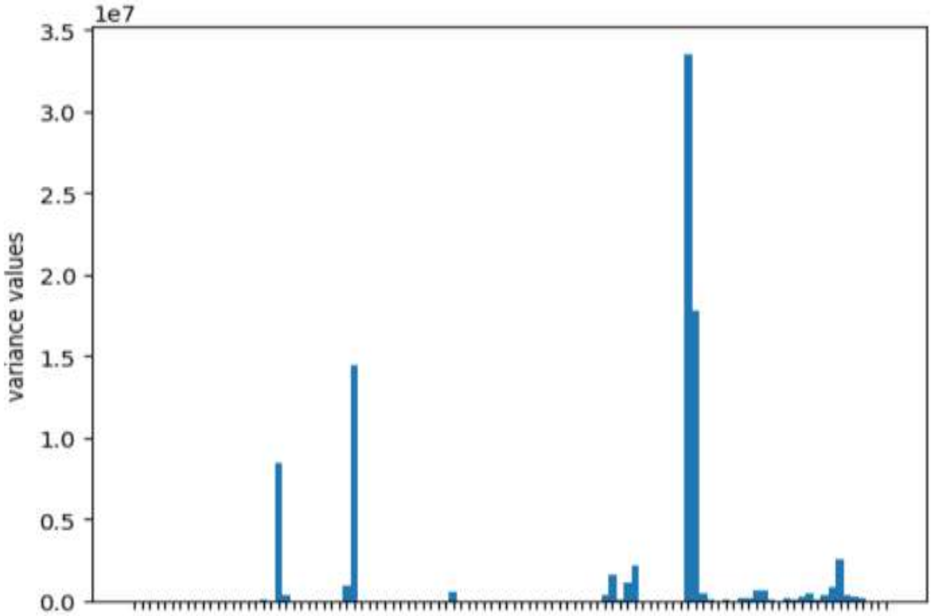

Fig 3.2 Variance distribution

F. Problem specific conditions

The model has unique challenges. Data distribution is such that direct normalization will result in loss of data. The probability of discriminator to oscillate less due to its target distribution which is a straight forward normal curve

$$y \sim N(\mu,\sigma)$$

μ is the mean value of the nutrition limits of food items.

IV. IMPLEMENTATION

A. Network architecture

Training diffusion based generation model:

- Determine initial levels of noise and data
- Repeat for significant steps:
    - Injecting noise into data
    - Train model to extract noise from noisy data
    - Increase the noise level
- Repeat above , until convergence

B. Training Model

Training the diffusion model to generate the data for random noise shows the following results
When the loss was found acceptable, the training was stopped and the model is saved .

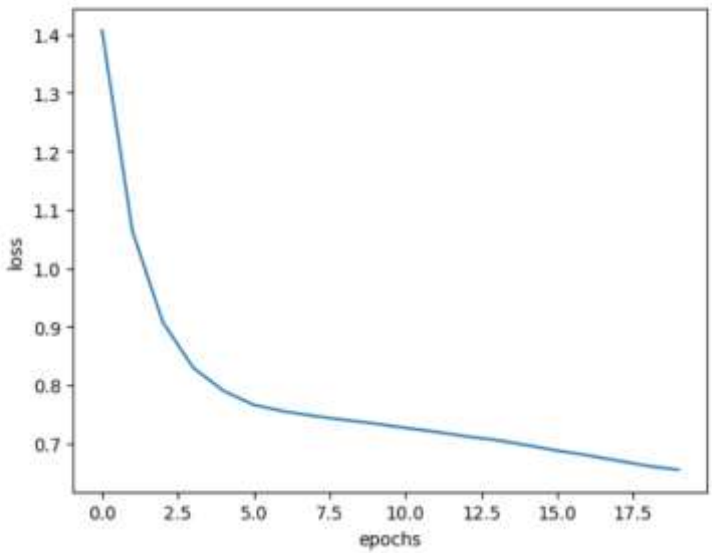

Fig 4.1 diffusion based generation model loss vs epochs

C. Training GAN:

GAN training cycle comprises of 7 epochs of generator with 5 epochs of discriminator.
For comparing the performance, GAN was trained with and without including the diffusion model in the training loop. The generator and discriminator networks are same in both cases.

Training the GAN directly on data failed to converge for lower loss. Adding block for diffusion, consumed more resources for computation but converged faster and settled at lesser loss.

Overall performance of model is evaluated based on the accuracy of generator to generate food items within the range and the distance of generated food items from its neighbor in the database calculated using k nearest neighbor.

D. Generator training for random training cycle

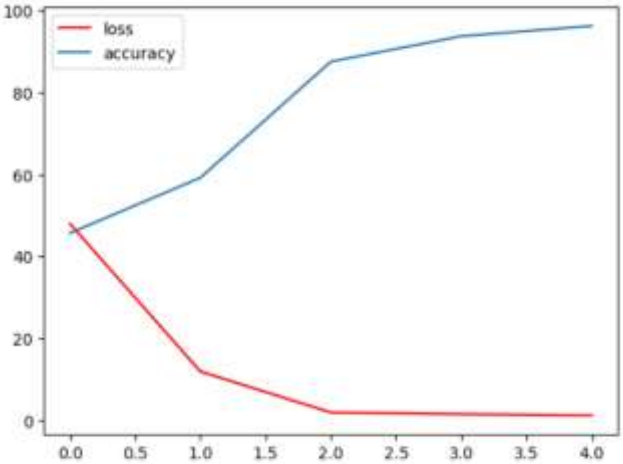

Fig 4.2 generator loss and accuracy vs epochs

E. Accuracy of generator and discriminator

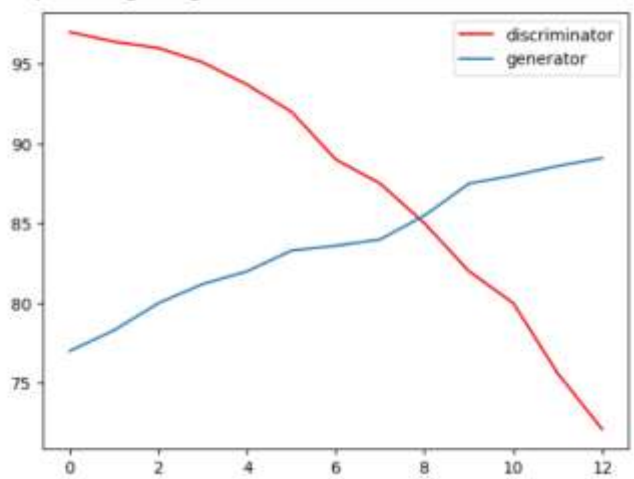

Fig 4.3 generator accuracy and discriminator accuracy vs epochs

F. Results and Analysis

	GAN training time	loss	accuracy	K nearest neighbour distance
GAN model	7 sec/ epoch for 100 training cycles	58	65	193
GAN model (with diffusion based generation block)	18 sec/epoch for 20 training cycles	0.23	90	15

Table 4.1 comparison table of GAN performance

GAN has been tested for faster convergence

V. LIMITATIONS AND FUTURE SCOPE

This paper is limited to the study of the effect of data on the training of the GAN. Various issues like mode collapse has not been addressed. Due to limited data available, Diffusion based generation model was used as a generalised data processor. But there are alternate ways of data processing which are application specific.

The objective of the model presented in the paper is to generated list of food items that meet the nutritional requirement of an average individual. This model can be further upgraded to generate meal plan given the age of the person by implementing a CGAN. Further this can be extended to incorporate functionality to select type of food and get personalised meal plan.

VI. ACKNOWLEDGMENT

I am thankful for data science professors of GEETAM Deemed University, Hyderabad for their patience and valuable support lasted throughout the work.

VII. REFERENCES

[1] Marwa Naili, Anja Habacha Chaibi, Henda Hajjami Ben Ghezala, Comparative study of word embedding methods in topic segmentation, Procedia Computer Science, Volume 112,2017, Pages 340-349, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2017.08.009.

[2] S. Bond-Taylor, A. Leach, Y. Long and C. G. Willcocks, "Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 11, pp. 7327-7347, 1 Nov. 2022, doi: 10.1109/TPAMI.2021.3116668.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, et al., "Generative adversarial nets", *Advances in neural information processing systems*, pp. 2672-2680, 2014.

[4] Q. Sun and W. Wang, "A Method to Accelerate the Training of WGAN," 2018 5th International Conference on Information Science and Control Engineering (ICISCE), Zhengzhou, China, 2018, pp. 51-54, doi: 10.1109/ICISCE.2018.00020.

[5] E. Levina and P. Bickel, "The Earth Mover's distance is the Mallows distance: some insights from statistics," Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vancouver, BC, Canada, 2001, pp. 251-256 vol.2, doi: 10.1109/ICCV.2001.937632.

[6] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, Yuichi Yoshida, spectral normalization for generative adversarial networks, ICLR 2018, arXiv:1701.07875

[7] Paulina Grnarova, Yannic Kilcher, Kfir Y. Levy, Aurelien Lucchi, Thomas Hofmann, Generative minimizing networks: Training GANs without compition, arXiv:2103.12685

[8] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, Ben Poole ,SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS, arXiv:2011.13456

[9] Martin Arjovsky , Soumith Chintala , and L´eon Bottou, Wasserstein GAN, arXiv:1701.0.7875.v3