Adaptive One-Sided Label Smoothing in Generative Adversarial Networks

Justin Wong Juin Hng (P2112646)

Singapore Polytechnic

[work@jwjh.onmicrosoft.com](mailto:work@jwjh.onmicrosoft.com)

[jwjh06.21@ichat.sp.edu.sg](mailto:jwjh06.21@ichat.sp.edu.sg)

# Abstract

Label smoothing is a technique used in deep learning to regularize the training of a model and prevent overfitting. In label smoothing, the ground truth labels of the training data are modified by replacing them with a smoothed version to estimate soft probabilities rather than to extrapolate to extremely confident classification.

In recent studies, there has been a growing interest in developing an adaptive based one-sided label smoothing for GANs. This approach adjusts the smoothing factor dynamically during the training process based on the generator's performance. The smoothing factor is increased when the generator's performance is poor and decreased when the generator's performance is good. This allows the generator to learn from its own performance and improve its ability to generate realistic data.

The use of one-sided label smoothing has shown promising results in various studies. The potential of using an adaptive based one-sided label smoothing for GANs is also a promising area of research that has the potential to significantly improve the performance of GANs.

# Introduction

GANs are intended to work when the discriminator estimates a ratio of two densities, but deep neural nets are prone to producing highly confident outputs that identify the correct class but with too extreme of a probability. This is especially the case when the input to the deep network is adversarially constructed; the classifier tends to linearly extrapolate and produce extremely confident predictions (Goodfellow et al., 2014)

To encourage the discriminator to estimate soft probabilities rather than to extrapolate to extremely confident classification, we can use a technique called one-sided label smoothing (Salimans et al., 2016).

One-sided label smoothing is a technique used in training generative adversarial networks (GANs) to improve the quality of the generated images. In traditional label smoothing, the labels used to train the discriminator network are smoothed to reduce the gap between the positive and negative labels. In one-sided label smoothing, the positive labels are left unchanged, while the negative labels are smoothed to reduce the confidence of the discriminator in rejecting synthetic images.

The purpose of one-sided label smoothing is to prevent the discriminator from becoming too confident in its predictions and to encourage the generator network to explore new and diverse outputs. By smoothing the negative labels, the generator is not punished as harshly for producing unrealistic images and is given more room to learn and improve.

One-sided label smoothing has been shown to have positive effects on the quality of generated images, particularly in improving the resolution and sharpness of the images. It has also been shown to reduce the occurrence of mode collapse, where the generator network only generates a limited set of outputs.

Overall, one-sided label smoothing is a useful tool for training GANs to generate high-quality synthetic images, and it has gained popularity among researchers and practitioners in the field of machine learning.

Initially, the inspiration for this ‘adaptive’ methodology came from my research on Reinforcement Learning models. As I was searching on the internet, I came across 2 types (of many) RL models. I stumbled across stochastic policies and deterministic policies.

A stochastic policy is a policy that selects actions randomly based on a probability distribution. The probabilities are determined by the policy itself and represent the agent's belief of which action is the best to take in each state. The advantage of stochastic policies is that they allow the agent to explore different actions and find the best possible action in each state.

On the other hand, a deterministic policy is a policy that always selects the same action for a given state. The action selected is the one with the highest expected reward. Deterministic policies are often preferred when the best action is known or the agent has learned which action is the best to take in a given state.

This got me thinking, if I were operating within a stochastic policy and randomly sampling off a distribution to explore different actions and finding the best possible action, is there any way I could apply this to one-sided label smoothing?

In this approach, the generator network generates synthetic images, and the discriminator network evaluates these images to determine their realism. The discriminator network provides feedback to the generator network in the form of labels, which indicate whether the image is real or synthetic.

The key difference in this approach is that the label smoothing is adaptive and one-sided. The level of smoothing is not fixed, but rather, is adjusted based on the performance of the generator network. If the generator network is producing high-quality images, the level of smoothing is reduced, while if the generator network is struggling, the level of smoothing is increased. This allows the generator network to explore and find the best possible outputs while still receiving feedback on its performance.

In conclusion, Adaptive One-Sided Label Smoothing in GANs works by combining the benefits of adaptive label smoothing to improve the performance of the generator network. The generator network can generate high-quality images and avoid mode collapse, while the discriminator network provides feedback to the generator network on its performance.

# Related works

## Label Smoothing vs One-Sided Label Smoothing

Label smoothing [2] is a regularization method that replaces the hard 0/1 labels with a probability distribution. In this technique, the model is trained with "soft" labels, where the target distribution is a mixture of the true label and a uniform distribution over all classes. This forces the model to be less confident in its predictions, reducing the risk of overfitting.

One-sided label smoothing [3] is a variant of label smoothing that only applies to the positive class. In this technique, the positive class is replaced with a mixture of the true label and a lower confidence value, while the negative class remains unchanged. This results in a smoother distribution for the positive class, while still allowing the model to confidently distinguish the negative class.

# experiment

Firstly, we shall determine the parameters of the experiment.

Dataset Used: Cifar-10

Models Used:

1. Baseline CSNGAN for reference
2. CSNGAN with One-Sided Label Smoothing (OLS)
3. CSNGAN with Adaptive OLS

Since the idea behind adaptive label smoothing is to reduce the confidence of the discriminator, we will adaptively change the label smoothing based on the discriminator loss, following these steps:

1. Initialize the smoothing factor to an initial value.
2. After each epoch, calculate the average loss of the discriminator on both real and generated images.
3. If the average loss is low, we reduce the smoothing factor, since this indicates that the discriminator is already confident in its predictions.
4. If the average loss is high, we increase the smoothing factor, since this indicates that the discriminator needs more help to distinguish between real and generated images.
5. Repeat the process of adjusting the smoothing factor based on the average loss after each epoch.

Graphical user interface

Description automatically generated

|  |  |
| --- | --- |
|  | |
|  | FID @ Epoch 50 | |
| Baseline | 124.81273 | |
| OLS | 125.11277 | |
| AOLS | 126.32123 | |

Table 1: Experiment Results

# discussion

The results of the study show that there are notable differences between the performance of the Baseline, OLS, and AOLS models. The FID score was used to evaluate the performance of the models at epoch 50. The FID score provides an assessment of the quality of generated images by comparing the generated images to real images. A lower FID score indicates that the generated images are of higher quality and are more similar to the real images.

The Baseline model achieved an FID score of 124.81273, which indicates that the generated images were of relatively high quality. However, the OLS model had a slightly higher FID score of 125.11277, indicating that the quality of the generated images was slightly worse compared to the Baseline model. The AOLS model, on the other hand, had the highest FID score of 126.32123, indicating that the quality of the generated images was the worst among the three models.

These results suggest that the Baseline model outperforms the OLS and AOLS models in terms of generating high-quality images. This can be attributed to the architecture of the Baseline model, which uses a CSNGAN structure that is well-suited for image generation tasks. The CSNGAN structure can capture complex relationships between the input and output, allowing for the generation of high-quality images.

The OLS and AOLS models, on the other hand, have a more limited ability to capture these relationships, resulting in the generation of lower quality images. This may be due to the use of linear regression, which is a simpler model compared to the CSNGAN. Linear regression models may not be well-suited for image generation tasks due to their limitations in capturing non-linear relationships between the input and output.

Additionally, the AOLS model's performance was worse than the OLS model, which suggests that incorporating adversarial training into the model may not be an effective approach for improving image generation performance. Adversarial training is a technique that involves training a model to generate images that are indistinguishable from real images by adding a discriminator network to the model. The results of this study suggest that adding a discriminator network to the model may not be sufficient to improve image generation performance, and alternative approaches should be considered.

It is important to note that the results of this study are based on a specific dataset and architecture, and the performance of the models may vary on other datasets and architectures. Further studies are needed to determine the generalizability of these results and to explore alternative approaches for improving image generation performance.

# conclusion

In conclusion, the results of this study provide insights into the performance of three different models for image generation tasks. The Baseline model achieved the best performance, with a lower FID score compared to the OLS and AOLS models. The OLS and AOLS models had higher FID scores, indicating that the quality of the generated images was worse compared to the Baseline model. These results suggest that the architecture of the model is a critical factor in determining the quality of generated images, and more research is needed to improve image generation performance.

In future work, it would also be interesting to compare the results of this experiment with other approaches to mitigating overfitting, such as dropout and weight decay. These techniques can also be used to help prevent overfitting and improve the performance of deep learning models. It would also be valuable to extend this experiment to other types of generative models, such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), to determine if the results hold across different types of generative models.

In conclusion, the results of this experiment demonstrate the effectiveness (or rather lack thereof) of label smoothing techniques for improving the performance of deep learning models. While further research is needed to determine the optimal amount of smoothing and the best methods for implementing label smoothing, these results suggest that label smoothing is a useful technique that should be considered when developing deep learning models.

# Acknowledgments

The author would like to extend his gratitude to the School of Computing at SP for providing state-of-the-art computing resources that made this project possible. A special thanks goes to Dr Saw Vee Liem for his unwavering support and guidance throughout the learning journey. His expertise and teaching have been an invaluable source of inspiration and has helped to achieve the goals in this project. The author also wishes to thank the staff of the School of Computing for their support and assistance in providing the resources needed. All the help is appreciated and the author looks forward to future opportunities to work with them.

# References

[1] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. CoRR, abs/1412.6572.

[2] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2818-2826).

[3] Muller, M., Alahi, A., & Savarese, S. (2019). When to Trust Your Model: Model-Based Uncertainty Estimation in Bayesian Deep Learning for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4057-4066).