Tuesday, 26. November 2024

Handout Kick-Off Meeting

Generative Data Augmentation using Multi-Agent Diverse Generative Adversarial Networks

Agenda

Research Topic

- Generative Data Augmentation (GDA)
- Vanilla GAN
- Multi-Agent Diverse GAN (MAD-GAN)
- Classifier with GDA & Research Questions

Challenges, Risks & Solutions

Technologies

Key Points Discussed

1. Generative Data Augmentation (GDA)

Definition:

Expanding training datasets by generating synthetic labeled samples to improve classification performance.

Applied in image, text, and audio domains.

Benefits:

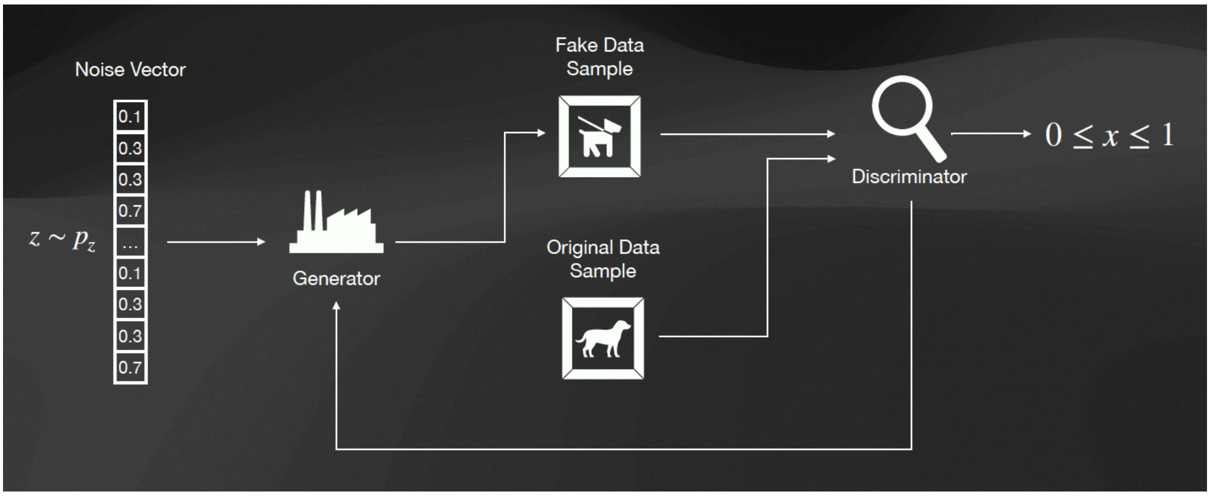
- Improves classification accuracy, especially with limited training data.
- Reduces overfitting, enhance generalization.
- Versatile across multiple scenarios.

Drawbacks:

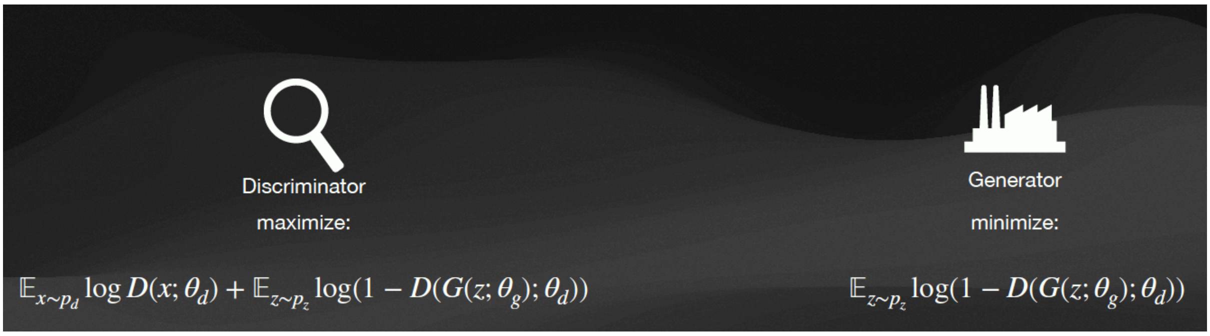
- May not be effective with insufficient training data.
- Requires manual tuning of augmentation datapoints.
- Can negatively impact performance with too limited training data to start with.

2. Vanilla GAN

Architecture:

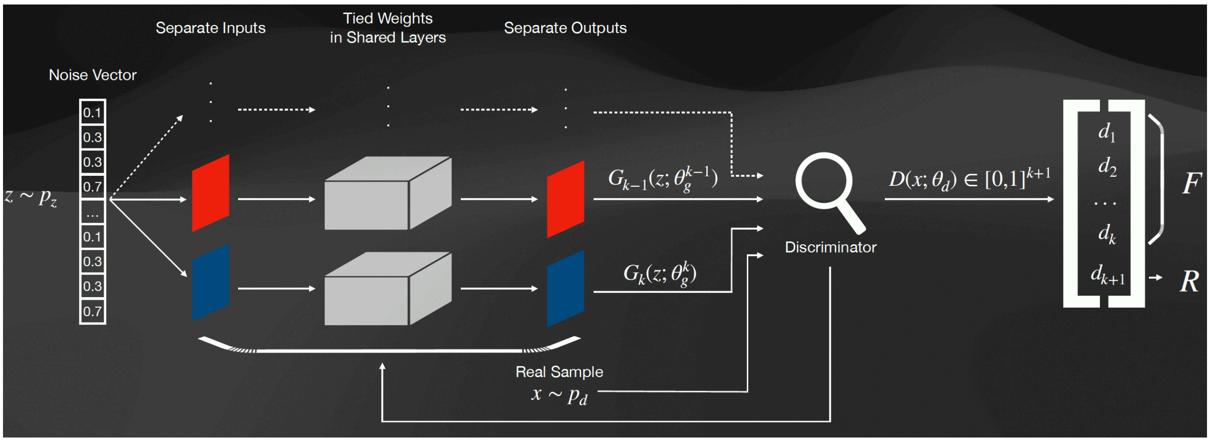


Objective functions:

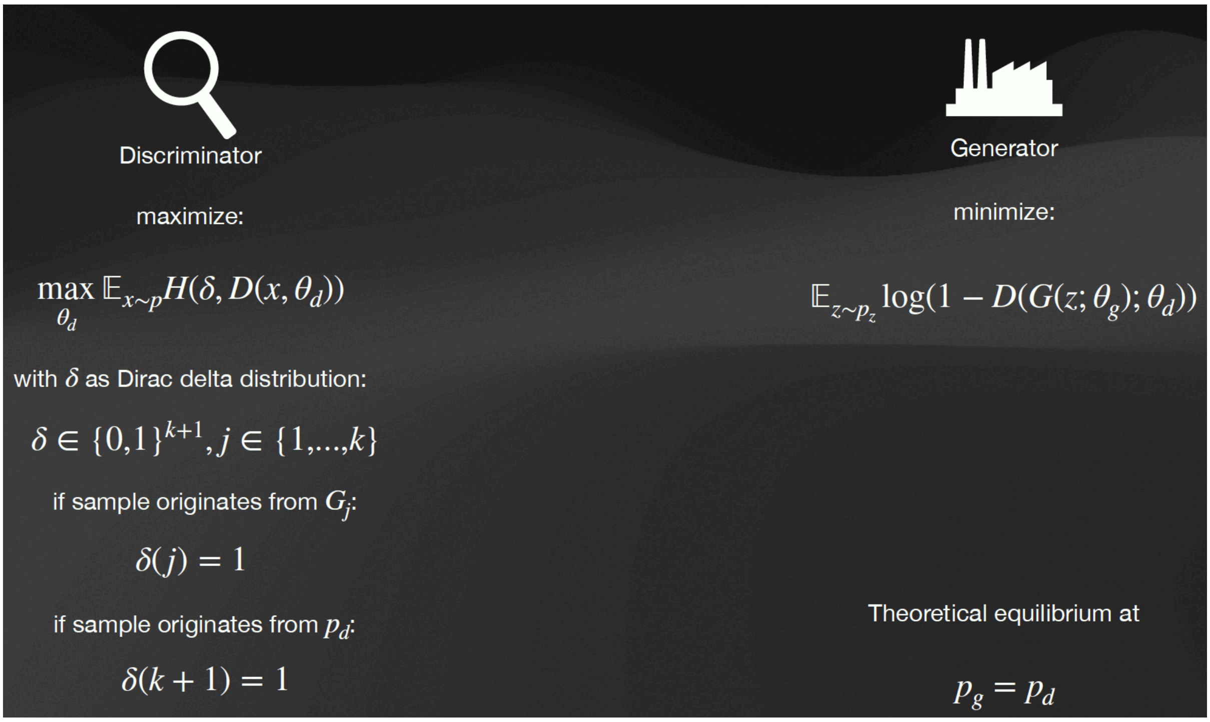


3. Multi-Agent Diverse GAN (MAD-GAN)

Architecture:



Objective functions:



4. Classifier with GDA

Points of research:

- Influence of MAD-GAN GDA on classifier performance.
- Impact of different loss functions on generated images.
- Effect of the number of generators in MAD-GAN.
- Ratio between real and fake samples for subsequent classifier.
- Comparison between MAD-GAN GDA, classic data augmentation, and Vanilla GAN GDA.

5. Challenges, Risks & Solutions

Challenges:

1. Combining MAD-GAN with conditionality constraint of CGANs.
2. High computational costs with high-definition training samples and high number of generators.
3. Sensitivity to hyperparameters.

Risks:

1. Incorrect combination of objective functions.
2. Generated images may degrade classifier performance.
3. Difficulty in proving statistical significance of improvements.

Solutions:

C/R1: Using auxiliary classifier and manual checking of generated images and corresponding classes.

C2: Use low-resolution datasets (e.g., MNIST, Fashion-MNIST, CIFAR-10/-100).

C3: Automated hyperparameter optimization and curriculum learning.

R2: Do not use to challenging dataset to limit the chances of occurrence.

R3: Perform statistical significance tests.

6. Technologies

Programming Libraries:

Anaconda, Python, TensorFlow, TensorFlow-Probability, Scikit-learn, Numpy, Pandas, Matplotlib, Seaborn, Jupyter, PIL.

Software:

Terminals, Jupyter Lab, VS Code, Tex Shop.

7. Sources

References to relevant literature and research papers for kick-off meeting:

Zheng et al. (2023), Azizi et al. (2023), Besnier et al. (2020), Goodfellow et al. (2014), Ghosh et al. (2018).