Paper

- **Title:** MGAN: Training Generative Adversarial Nets With Multiple Generators
- **Authors:** Quan Hoang, Tu Dinh Nguyen, Trung Le, Dinh Phung
- **arXiv link:** https://arxiv.org/abs/1708.02556

TL;DR

The paper shows how to use multiple Generators to train a GAN to avoid mode collapse. The idea is to make the mixture of all generators learn the real distribution while being as distinct from each other as possible i.e. minimize the Jenson Shannon divergence (JSD) between the real data distribution and the mixture of distributions learned by the generators while maximizing the JSD among the distributions learned by individual generators.

Method

- There are K generators $G_{1:K}$, a discriminator D and a classifier C
- Each generator maps z to x, learning a distribution P_{G_k} and all K generators learn the mixture distribution P_{model}
- An index u is drawn from a fixed multinomial distribution and then sample $G_u(z)$ is used as the output from P_{model}
- Discriminator D tries to distinguish between this sample and sample from the real data (P_{data})
- The classifier C does multi-class classification to classify samples labeled by the index of the generating generator.

The Game

The following multi-player minimax game is being played.

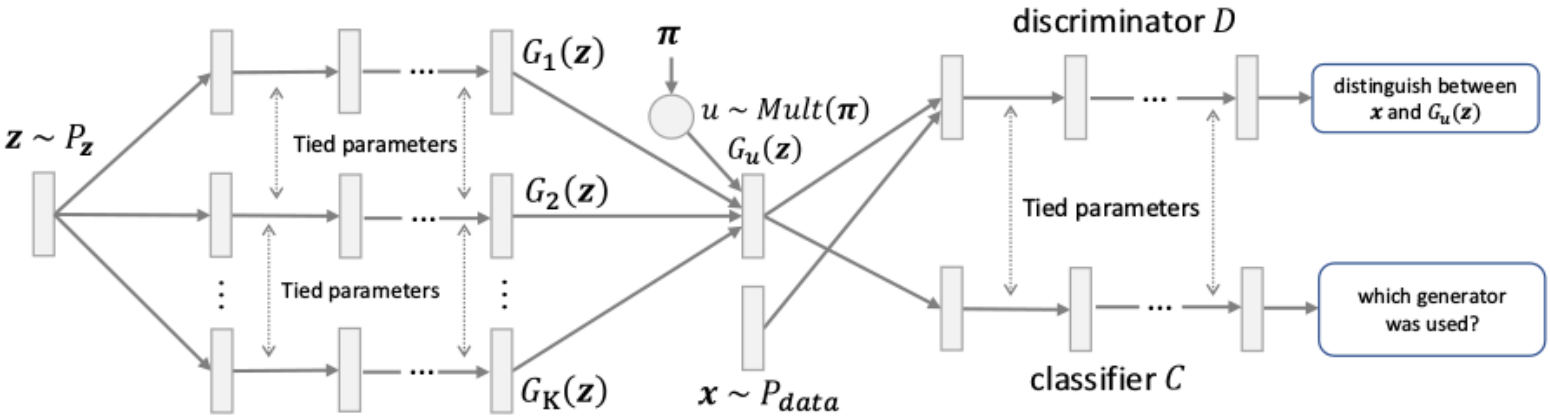
$$\min_{G_{1:K},C} \max_{D} J(G_{1:K}, C, D) = \mathbb{E}_{x\sim P_{data}}[log\ D(x)] + \mathbb{E}_{x\sim P_{model}}[log\ (1-D(x))] - \beta \left\{ \sum_{k=1}^{K} \pi_k \mathbb{E}_{x\sim P_{G_k}}[log\ C_k(x)] \right\}$$

where $C_k(x)$ is the probability that the classifier assigns to x being generated by generator k and $\beta > 0$ is the diversity parameter.
The description of the minimax game is:

- The Discriminator aims to distinguish between real data and data generated by the generator

- The Classifier aims to classify which generator generated the data
- Each of the K generators aim to fool the discriminator into classifying it's generated data as real while also assisting the classifier in identifying the correct label of the index of the generator.

Architecture



Parameter sharing is used to reduce the number of parameters. Only the parameters in the first layer of generator and last layer of discriminator are not shared.