Generative Adversarial Networks Image Generation

Generative Adversarial Networks are a type of machine learning model that can create new, realistic data, like images or audio, based on patterns from real examples. They're made up of two parts: a generator that creates fake data, and a discriminator that tries to tell if the data is real or fake. These two compete with each other; the generator keeps trying to get better at fooling the discriminator, and the discriminator keeps trying to catch the fakes. This back-and-forth helps the generator improve until it gets really good at producing data that looks real.

For this experiment, I used the pretrained BigGAN model (biggan-deep-256) to explore how random noise can be turned into realistic-looking images. After loading the model, I provided it with a fixed class vector representing a golden retriever, along with a truncation value of 0.4, which helps balance image realism and variety. I then generated a 128-dimensional random latent vector using torch.randn, which acts as the input noise for the generator. Passing this noise, along with the class vector and truncation value, into the model produced an image. To experiment further, I regenerated new latent vectors multiple times to see how different random inputs would lead to different variations of the same class.

Out of the five images I generated, most of them looked like golden retriever dogs; they had features like paws, noses, and fur, usually in a similar brown color. One of the images wasn't really recognizable, and the dog looked like it was mixed with a person. Overall, the dogs weren't perfectly shaped; their ears were sometimes well-formed, but their noses and eyes were all distorted. In one case, the dog's face seemed to be replaced with the face of a different breed.

I learned that GANs use random numbers to control different features in the images they create. Each number in the noise vector affects something specific, like hair length, face shape, or eye position. Since the values are randomly picked each time, the model ends up generating a different-looking person with every new input.

One area that could be improved is the model's training on the placement of facial features, especially the eyes. It's possible that there aren't enough detailed examples in the training data for this specific feature. While some eye shapes and details look somewhat accurate, others resemble features from different breeds, like huskies. In some cases, certain features are close but repeated multiple times, for example, one image had three noses. The main challenge seems to be getting the model to accurately place each detail the correct number of times, with better precision and consistency.

https://github.com/castilloadriana/Cognizant-GenAI-Externship-/blob/main/Assignments/gan_assignment.ipynb

Generative Adversarial Networks Image Generation