A Style-Based Generator Architecture for Generative Adversarial Networks

Tero Karras, Samuli Laine, Timo Aila

https://arxiv.org/abs/1812.04948
https://arxiv.org/pdf/1812.04948.pdf

TL;DR

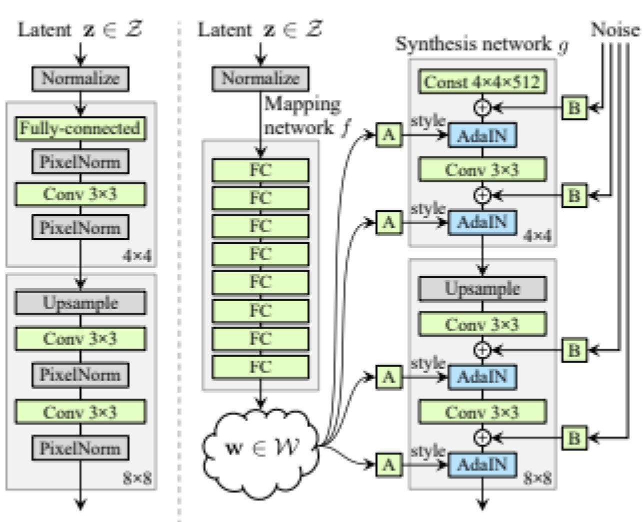
An alternative generator architecture for GANs, borrowing from style transfer literature. The architecture leads to automatically learned separation of high level attributes (pose, identity) and stochastic variation (freckles, hair), and enables intuitive, scale-specific control of the synthesis. The new generator improves quality, has better interpolation properties, and can better disentable latent factors. A new high-quality dataset of human faces is also introduced.

Introduction

Other GANs operate as a blackbox, and the properties of the latent space are poorly understood. The new generator architecture exposes novel ways to control the image synthesis. It starts from a learned constant input and adjusts the “style” of the image at each convolution layer based on the latent code, and can thus control the strength of image features at different scales. The discriminator and loss function remains the same. The generator first embeds the input latent code into an intermediate latent space. This allows the latent space to be disentangled as it does not follow the same probability density of the training data. For example, if a dataset doesn’t contain any images of long haired males, the generator would correlate long hair and females, and be unable to generate long haired males. The mapping network disentangles these correlations.

Style-based generator

Traditionally the latent code is provided to the generator through an input layer. The new architecture omits this layer and starts from a learned constant. A non-linear mapping network maps the input latent code into an intermediate latent space *W*.



(a) Traditional (b) Style-based generator

Learned affine transformations then transform the intermediate latent space to styles $y=(y_s, y_b)$ that control adaptive instance normalization (AdaIN) operations that are applied after each convolution layer.

$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i},$$

x_i is the feature map. Random gaussian noise is also introduced to the system, to provide the generator with means to generate stochastic detail.

Properties of the style-based generator

The generator architecture makes it possible to control the image synthesis via scale-specific modifications to the styles. The mapping network and affine transformations can be viewed as a way to draw samples for each style from a learned distribution, and the synthesis network as a way to generate a novel image based on a collection of styles.

This network also allows for stochastic variation like the exact placement of hairs.

(a) Generated image (b) Stochastic variation

Disentanglement

Two methods of quantifying disentanglement without the use of an encoder:

1. Measure the difference between two VGG16 embeddings when interpolating between two inputs.
2. Measure how well latent-space points can be separated into two distinct sets, so that each set corresponds to a specific binary attribute of the image. (e.g. male and female). The easier the latent space is separable, the more separable the features.

Conclusion

Traditional GAN generator architectures are vastly inferior to style-based designs.