

Summary/Review

Variational Autoencoders

Variational autoencoders also generate a latent representation and then use this representation to generate new samples (i.e. images).

These are some important features of variational autoencoders:

- Data are assumed to be represented by a set of normally-distributed latent factors.
- The encoder generates parameters of these distributions, namely μ and σ .
- Images can be generated by sampling from these distributions.

VAE goals

The main goal of VAEs: generate images using the decoder

The secondary goal is to have similar images be close together in latent space

Loss Function of Variational Autoencoders

The VAE reconstruct the original images from the space of vectors drawn from a standard normal distribution.

The two components of the loss function are:

- A penalty for not reconstructing the image correctly.
- A penalty for generating vectors of parameters μ and σ that are different than 0 and 1, respectively: the parameters of the standard normal distribution.

Generative Adversarial Networks (GANs)

The invention of GANs was connected to neural networks' vulnerability to adversarial examples. Researchers were going to run a speech synthesis contest, to see which neural network could generate the most realistic-sounding speech.

A neural network - the "discriminator" - would judge whether the speech was real or not.

In the end, they decided not to run the contest, because they realized people would generate speech to fool this particular network, rather than actually generating realistic speech.

These are the step to train GANs:

- Randomly initialize weights of generator and discriminator networks

- Randomly initialize noise vector and generate image using generator
- Predict probability generated image is real using discriminator
- Compute losses both assuming the image was fake and assuming it was real
- Train the discriminator to output whether the image is fake
- Compute the penalty for the discriminator probability, without using it to train the discriminator
- Train the generator to generate images that the discriminator thinks are real
- Use the discriminator to calculate the probability that a real image is real
- Use L to train the discriminator to output 1 when it sees real images