AI/ML Notes - Stable Diffusion Paper - 11/2/2022

Diffusion is a way to generate data from the internal structure of the image instead of generating it from explicit labels.

Deep learning with diffusion models - it’s easy to fit things to Gaussian models, easy to parameterize and fit data, but not very descriptive. The idea is to have a series of steps that turn distributions into Gaussians. Markov kernels take an initial distribution and multiplies it with a Gaussian. This is then discretized into a series of steps - you form a kernel, then progress all the way to the timestep where you have a normal distribution. You can now take a prior normal distribution and apply the reverse, these Markov distributions to produce the original distribution. Distribution is shown as integral, like a convolution moving toward infinity. Each stage is applying T (Markov kernel) to each term before it.

The kernel is iteratively gathering information about the image from random noise.

Question: So for each image you’ve got a similar process to get random noise - you define a forward process which is doing the little blurry bits, but you’re also learning the reverse of this process timestep by timestep.

TJ: another paper models this diffusion process as a stochastic differential equation.

Up until a few years ago, the predominant way of generating images was GAN, where you have a generative model that produces data from random noise and is then trained based on a discriminator network trained on real data. Recently, with VQGANs, they learned to represent the changes the image was going through from random noise and encode this in a latent space.

Then:

<https://arxiv.org/abs/2105.05233>

The way diffusion is done is through a denoising U-Net that predicts the noise at each phase.

They formulate the original loss as the difference between the noise predicted and the noise added. You actually go from Zt to Zt-1 by subtracting the output of the U-Net.

Question about hierarchical U-Net: Matt: Refers to explicit spatial hierarchy in U-Net.

Transformer encoder - each output gets snapped to a particular point in space, recorded as vectors in a codebook.

Perceptual compression - it compresses the space, e.g. jpeg

Semantic compression - you compress the image, lose features of the image, and can create similar image but you are doing it in a way where you’re learning a much denser form of the image and you’re representing that the image is e.g. a corgi vs. showing every original detail.

This U-Net itself is a full neural network and it’s not computationally cheap.

Why are we using it in the first place? It’s much more generalizable. In the end we’re working with spatial features, so you can be slightly off on certain details but you know a table has to have legs beneath a flat surface. Training on the latent space is cheaper but much better than just fitting a model to a set of vectors. Now that you have the latent space, how do you decode it? One way is through vector quantization, fitting to a codebook of vectors to generate the image. The other is using KL divergence which involves modeling the difference between the generated images and the expected images.

The inverse of distribution is still a distribution, due to the result of denoising it could be many things, so it is a vector choses from a distribution or KL divergence. (KL divergence is not just a likelihood or a distance but an expectation of a certain input distribution. If you want to determine the difference between two distributions, you use KL divergence).

How to factor in image captions: conditioning mechanisms

In the U-Net, they integrated attention. You have three different inputs to transformer: Query, Key, Value. It takes the similarity between Q and K and maps it to V. The Q is the noised input K and V are the outputs of the encoded condition. Each convolution of the U-Net is called a cross-attention because text and image encoding are different spaces, so this represents attention connecting different spaces.

The K and V are not exactly the outputs of the conditioning, they go through a linear translation first. The attention model can determine which is the best space for K and V areas to be mapped to.

Layout-to-image synthesis - based on image regions / semantic maps.

Matt: how does the schema make it into the layout-to-image synthesis? More exploration, not yet conclusive. How is tau theta in appendix done? P. 25-27.

…

Metric for evaluating generative models:

<https://en.wikipedia.org/wiki/Fr%C3%A9chet_inception_distance>

Inception uses residual layers within its architecture.