DreamBooth loss

Filip Strand

2024-10-27

Given a reference image X we can encode it using the `vae` to get l_T. Thus, l_T represents a (clean) latent without any added noise. In this derivation, we use the notation from `MFLUX`[1] and let t move from $t = 0$ to $t = T$ when generating an image from noise. In other words, when generating an image from pure noise, we sample a $l_0 \sim \mathcal{N}(0, 1)$ (a pure noise latent) and progressively move to l_T, the clean image latent, (which would subsequently be passed to the `vae` for decoding)

$$l_0 \to l_1 \to l_2 \to ... \to l_T$$

The time steps $t = 0, t = 1, t = 2, \dots$ etc. have a direct correspondence with a set of σ values $\sigma_0, \sigma_1, \dots$. While t progresses in discrete integer steps $0, 1, 2, \dots,$ the σ values are floats that progressively go down in value from 1.0 to 0.0. As an example, it could look like the following:

$$t_0 = 0, t_1 = 1, t_2 = 2, \dots t = T$$
$$\sigma_0 = 1.0, \sigma_1 = 0.96, \sigma_2 = 0.92, \dots \sigma_T = 0.0$$

The exact value of the σ_t depend on how many time steps T we use.

Given this, we can get the equivalent noised version of the clean l_T at a time step t by linearly interpolating between a pure noise array ϵ (same thing as l_0) and the clean latent l_T:

$$l_t = (1 - \sigma_t) \cdot l_T + \sigma_t \epsilon$$

As can be seen, when $t = 0$, i.e $\sigma_0 = 1.0$ then this gives

[1]Which might be contrary to standard terminology

$$l_0 = (1 - \sigma_0) \cdot l_T + \sigma_0 \epsilon = \epsilon$$

and, conversely, when $t = T$, i.e $\sigma_T = 0.0$ we get

$$l_T = (1 - \sigma_T) \cdot l_T + \sigma_T \epsilon = l_T$$

Suppose we create two subsequent latents l_t and l_{t+1} by the linear interpolation method above:

$$l_t = (1 - \sigma_t) \cdot l_T + \sigma_t \epsilon$$
$$l_{t+1} = (1 - \sigma_{t+1}) \cdot l_T + \sigma_{t+1} \epsilon$$

As noted above, since the transformer part of the diffusion model (denoted T_θ here, with tunable parameters θ) takes us from a latent l_t to l_{t+1}, we can feed it l_t and the explicit time step t and get out ϵ'_t, for the predicted noise array

$$\epsilon'_t = T_\theta(l_t, t)$$

Then, we get the predicted next latent l'_{t+1} by taking a dt_t size step

$$l'_{t+1} = l_t + \epsilon'_t \cdot dt_t = l_t + \epsilon'_t \cdot (\sigma_{t+1} - \sigma_t)$$

where $dt_t = (\sigma_{t+1} - \sigma_t)$. The idea now is to form the loss by trying to get l'_{t+1} close to l_{t+1}. Lets write this, somewhat informally, as

$$\boxed{l'_{t+1} \approx l_{t+1}}$$

Then we can start substituting in our expressions for the LHS and RHS to simplify this expression. Starting by replacing l'_{t+1} on the the LHS, we get:

$$l_t + \epsilon'_t \cdot (\sigma_{t+1} - \sigma_t) \approx l_{t+1}$$

Now replacing l_{t+1} on the RHS we get

$$l_t + \epsilon'_t \cdot (\sigma_{t+1} - \sigma_t) \approx (1 - \sigma_{t+1}) \cdot l_T + \sigma_{t+1} \epsilon$$

Solving for our predicted ϵ'_t on the left hand side, we get

$$\epsilon'_t \approx \frac{(1 - \sigma_{t+1}) \cdot l_T + \sigma_{t+1}\epsilon - l_t}{(\sigma_{t+1} - \sigma_t)}$$

Further substituting in the expression for l_t that we got from linear interpolation we get

$$\epsilon'_t \approx \frac{(1 - \sigma_{t+1}) \cdot l_T + \sigma_{t+1}\epsilon - ((1 - \sigma_t) \cdot l_T + \sigma_t\epsilon)}{(\sigma_{t+1} - \sigma_t)}$$

Simplifying the numerator by collecting ϵ and l_T terms we get

$$\epsilon'_t \approx \frac{(\sigma_{t+1} - \sigma_t) \cdot \epsilon - (\sigma_{t+1} - \sigma_t) \cdot l_T}{(\sigma_{t+1} - \sigma_t)}$$

Finally, we see that the common factor of $(\sigma_{t+1} - \sigma_t)$ cancels and we get

$$\epsilon'_t \approx \epsilon - l_T$$

In order to turn this into an optimization objective we simply take LHS minus RHS, squared, and try to make sure this difference goes to zero. This means that our final loss can be written as

$$\boxed{\texttt{loss} := ||\epsilon'_t - (\epsilon - l_T)||^2}$$

or, equivalently,

$$\boxed{\texttt{loss} := ||\epsilon'_t + l_T - \epsilon||^2}$$

or, by emphasizing the transformer part of the network,

$$\boxed{\texttt{loss} := ||T_\theta(l_t, t) + l_T - \epsilon||^2}$$

In this form, the loss can be interpreted as having a the clean image latent l_T, then at an arbitrary time step t, we let the transformer predict the noise and the resulting quantity should be close to ϵ.

One nice thing to note here is that this loss holds for arbitrary t values within our range $[0, T)$. This property is computationally very efficient since we don't need to compute the full chain of diffusion steps during training.