Application Log 2

Hanzhang Yin

September 6, 2024

[1] **zhang2023addingconditionalcontroltexttoimage**.

Background Summary

ControlNet introduces a novel neural network architecture that integrates spatial conditioning controls into existing large-scale text-to-image diffusion models. Utilizing a robust backbone built on pretrained encoding layers and innovative ”zero convolutions,” ControlNet enhances model adaptability to various conditioning controls like edges and depth, tested across datasets of varying sizes. The results demonstrate ControlNet's potential to broaden the application scope of image diffusion models through precise control and fine-tuning capabilities.

Mathematical Content

The mathematical formulation of ControlNet modifies the Stable Diffusion (SD) model to incorporate additional conditions for controlling the image generation process. The key concepts and steps are outlined as follows:

Stable Diffusion Model without ControlNet

The original image generation process using the Stable Diffusion model can be expressed as:

$$\hat{x} = f_\theta(x, t, c)$$

where: $\hat{x}$ is the generated image; f_θ is the Stable Diffusion model with parameters θ;

x is the input image; t represents the time step; c is the conditional information (e.g., text prompt).

Stable Diffusion Model with ControlNet

After integrating ControlNet, the image generation process becomes:

$$\hat{x} = f_\theta(x, t, c, c') + \mathcal{T}_\phi(c'')$$

where: c' is the additional control condition (e.g., edge map, depth map), $\mathcal{T}_\phi$ represents the zero convolution modules with parameters ϕ; c'' is the transformed control condition in the latent space.

Zero Convolution Modules Initialization

Zero convolution modules are initialized with:

$$w_{\text{zero}} = 0, \quad b_{\text{zero}} = 0$$

where: w_{zero} and b_{zero} are the weights and biases of the zero convolution layers, both set to zero initially.
The convolution operation for a general convolutional layer can be represented as:

$$y_{i,j,k} = \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{c=1}^{C} x_{i+m,j+n,c} \cdot w_{m,n,c,k} + b_k$$

where: $y_{i,j,k}$ is the output feature map at spatial location (i, j) and channel k; $x_{i+m,j+n,c}$ is the input feature map at spatial location $(i+m, j+n)$ and channel c; $w_{m,n,c,k}$ are the weights of the convolutional filter of size $M \times N$; where c is the input channel and k is the output channel; b_k is the bias term for the output channel k, $M \times N$ is the size of the convolutional kernel (filter); C is the number of input channels.

Gradient Descent and Weight Update

During training, the gradient descent updates the weights, ensuring they become non-zero:

$$\text{If } w_{\text{zero}} = 0, \text{ and } \frac{\partial \mathcal{L}}{\partial w_{\text{zero}}} \neq 0, \text{ then after one step, } w_{\text{zero}} \neq 0$$

where:

- $\mathcal{L}$ is the loss function being minimized.
- $\frac{\partial \mathcal{L}}{\partial w_{\text{zero}}}$ is the gradient of the loss with respect to the zero convolution weights.

5. Joint Modeling of ControlNet

ControlNet jointly models the image and pixel-level conditions by modifying the loss function:

$$\mathcal{L}_\theta(x, t, c, c') = E_{x \sim p(x)} \left[\| f_\theta(x, t, c, c') + \mathcal{T}_\phi(c'') - x \|^2 \right]$$

where:

- E denotes the expected value over the data distribution $p(x)$.
- The loss function $\mathcal{L}_\theta$ aims to minimize the difference between the generated image and the target image.