**Intuition**

NN learning to remove noise and turn it back to image.

A screenshot of a computer game

Description automatically generated

**Sampling**

**Diffusion model 中timestamps 的作用是什么**

在扩散模型（Diffusion Model）中，timestamps（时间戳）是一个重要的参数，它用于控制扩散过程的时间步骤。时间戳在扩散模型中的作用是确定在每个时间步骤中如何操作噪声以逐渐生成更清晰的数据或图像。

具体来说，时间戳在扩散模型中的作用如下：

* 控制噪声水平：时间戳决定了噪声水平的强度。在扩散模型中，噪声通常被视为随时间递减的过程。初始时间步骤具有较高的噪声水平，而随着时间的推移，噪声逐渐减小，使数据或图像变得更清晰。
* 生成渐进性的图像：通过在每个时间步骤中逐渐减小噪声，扩散模型可以生成渐进性的图像或数据。开始时，生成的内容可能模糊或有噪点，但随着时间步骤的增加，图像逐渐变得更加清晰和真实。
* 控制生成质量：时间戳也可以用于控制生成的质量。通过增加时间步骤的数量或改变时间戳的分布，您可以调整生成图像的质量和细节水平。更多的时间步骤通常会产生更高质量的结果，但也需要更多的计算资源。
* 适应不同数据类型：时间戳的设置可以根据不同的数据类型和应用场景进行调整。对于不同的任务，您可以选择不同的时间戳策略，以满足特定的生成需求。

**Neural Network(U-Net)**

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

**Controlling with Embeddings**

Contexts are one-hot encoding.

A screenshot of a game

Description automatically generated

**Speed up**

DDIM: Denoising Diffusion Implicit Models

# fast sampling algorithm with context

@torch.no\_grad()

def sample\_ddim\_context(n\_sample, context, n=20):

# skipping n timestamps

# x\_T ~ N(0, 1), sample initial noise

samples = torch.randn(n\_sample, 3, height, height).to(device)

# array to keep track of generated steps for plotting

intermediate = []

step\_size = timesteps // n

for i in range(timesteps, 0, -step\_size):

print(f'sampling timestep {i:3d}', end='\r')

# reshape time tensor

t = torch.tensor([i / timesteps])[:, None, None, None].to(device)

eps = nn\_model(samples, t, c=context) # predict noise e\_(x\_t,t)

samples = denoise\_ddim(samples, i, i - step\_size, eps)

intermediate.append(samples.detach().cpu().numpy())

intermediate = np.stack(intermediate)

return samples, intermediate

**Deterministic Inference（确定性推断）：** DDIM采用了确定性的推断方法，即在每个时间步上，它不对噪声的概率分布进行建模，而是直接预测和应用噪声样本。相比之下，DDPM在每个时间步上都要对噪声的概率分布进行建模，并执行随机采样。这种确定性推断方法可以加速模型的训练和采样过程。

**快速噪声预测模型**： DDIM通常使用神经网络来预测噪声，这个网络可以被设计成较轻量级的结构，以便快速生成噪声。相比之下，DDPM可能需要更复杂的概率模型来建模噪声分布，这可能导致更高的计算成本。

**近似方法**： DDIM在一些情况下可能采用一些近似方法来简化计算，从而加速训练和采样过程。这些近似方法可能会引入一些误差，但通常可以在保持生成质量的前提下显著降低计算成本。