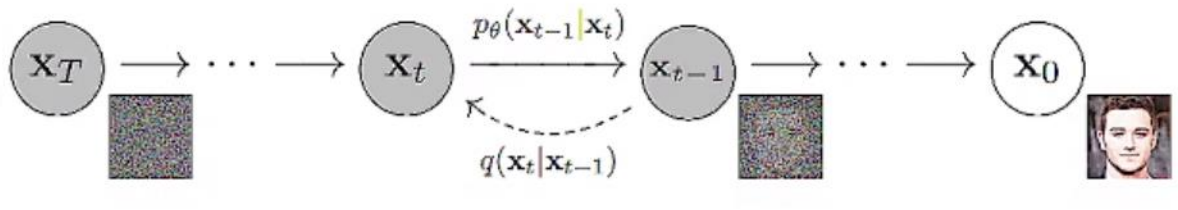
Diffusion Model (DDPM, DDIM)



Forward process(q)는 입력 영상에 노이즈를 서서히 추가하고, Reverse process(p_theta)는 노이즈를 서서히 제거하여 다시 원본 영상을 만드는 생성 모델이다. VAE 와 같은 기존의 생성 모델과는 다르게 여러 step 을 거쳐 이미지를 생성하는 것이 특징이다.

Diffusion Model 은 마르코프 성질을 가정한다. 마르코프 성질은 미래 상태의 확률은 현재 상태에 의해서만 결정된다는 것으로 과거 상태와는 독립적임을 나타낸다. 즉 forward 과정에서 t 번째 step 의 분포는 $q(x_t|x_0,x_1,x_2,...,x_{t-1})$로 나타낼 필요가 없고, $q(x_t|x_{t-1})$만으로 표현할 수 있다.

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) \coloneqq \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

위 수식은 Conditional gaussian distribution 으로, 어떤 분포(X_{t-1})에 표준 가우시안 분포를 더한 분포를 의미한다. B_t 는 가우시안 노이즈를 부여하는 정도로, 0.0001 과 같은 매우 작은 수라고 생각하면 되고 실제로 노이즈를 더하는 과정은 아래의 수식과 같다. 선형보간법을 이용하여 원본 영상은 감쇠하고, 감쇠된 만큼 노이즈를 더하는 것이다. 이 때, sqrt(B_t)는 매우 작은 수부터 시작되므로 이미지는 매우 서서히 노이즈가 되어간다.

X_t = sqrt(1-B_t)*X_{t-1}+ sqrt(B_t)*gaussian noise (=)

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon} \quad \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$$

근데 위 수식을 자세히 보면, noise 에 forward 분포에 대한 표준편차를 곱하고 평균을 더하는 과정이다. 이것은 VAE 에 나왔던 Reparameterization trick 과 동일하다. VAE 에서는 표준정규분포에서 샘플링한 값을 통해 정규분포에서 샘플링한 값을 계산하는 용도였다면, DDPM 에서는 노이즈가 부여된 영상을 만드는 용도로 이용될 수 있다.

즉 Conditional gaussian distribution 은 (노이즈, 평균, 표준편차)를 알면 해당 분포에 해당하는 이미지를 만들 수 있다.

다음은 실제 코드이다.

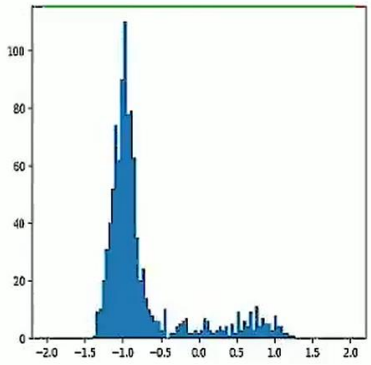
```
beta_s = 0.0001
beta_e = 0.01
timestep_num = 1000
```

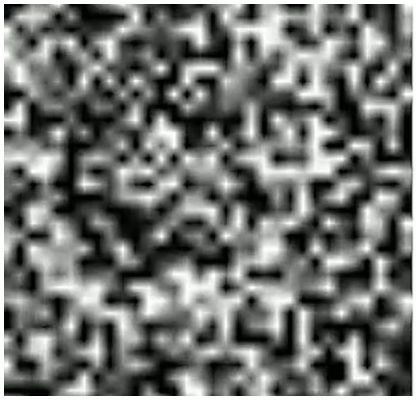
```
beta_list = np.linspace(beta_s,beta_e,timestep_num,dtype=np.float64)
```

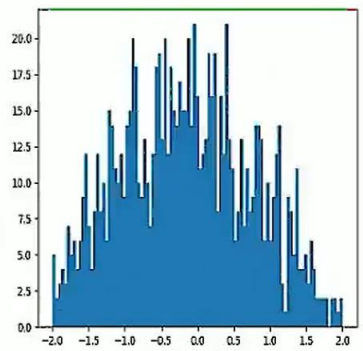
```
alpha_list = 1.0 - beta_list
```

```
esp = np.random.normal(0,1,(32,32))
img = np.sqrt(alpha_list[t]) * img + np.sqrt(beta_list[t]) * esp
```

노이즈가 추가될수록(t step 이 증가할수록) 분포는 점점 표준 가우시안으로 되어간다.







노이즈를 추가할 때 t step 을 0 부터 1000 까지 차례대로 줄 수도 있겠지만 원본 이미지를 통해 한 번에 노이즈를 추가할 수도 있다. 해당 방식은 아래의 수식과 같은데, $q(X_t|X_{t-1})$에서 평균과 분산을 a 로 표현하고 a 위에 bar 붙인 형태로 정리된다고 한다. 이 때, 노이즈 이미지는 Reparameterization trick 을 이용하였다.

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$\alpha_t := 1 - \beta_t \text{ and } \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$$

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0$$

위의 내용은 아래의 코드를 통해 구현할 수 있다.

```
alpha_bar_list = np.cumprod(alpha_list)
```

```
sqrt_alpha_bar_list = np.sqrt(alpha_bar_list)
```

```
sqrt_one_minus_alpha_bar_list = np.sqrt((1.0-alpha_bar_list))
```

```
img_show = sqrt_alpha_bar_list[t] * img + sqrt_one_minus_alpha_bar_list[t] * esp
```

학습은 Forward process 의 분포를 이용하여, Reverse process 에서 t 번째 이미지가 주어졌을 때 t-1 번째 이미지를 예측하는 것을 목표로 한다. 이것은 $p(x_{t-1}|x_t)$의 이미지 분포를 $q(x_{t-1}|x_t)$의 이미지 분포에 근사하는 과정으로 해결할 수 있다. 이 때, Forward process 에서 x_0 는 항상 알고 있기 때문에 condition 을 준 식인 $q(x_{t-1}|x_t,x_0)$로 표현해도 무방하다. 이후 $q(x_{t-1}|x_t,x_0)$와 $p(x_{t-1}|x_t)$의 분포가 유사해지도록 KLD 를 통해 최적화하였다.

$$\sum_{t>1} \underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}$$

논문의 저자들은 q 의 분포를 컨디셔널 가우시안으로 정의하였고, 가우시안 분포는 평균과 분산이 가장 중요하다. (평균과 분산만으로 분포를 표현할 수 있기 때문) 그리고 앞에서 언급했듯이, 평균과 분산을 통해 Reparameterization trick 을 이용해서 이미지로의 변환도 가능하다.

그래서 저자들은 먼저 $q(x_{t-1}|x_t,x_0)$의 평균과 분산을 베이지안 룰로 계산하였다. 변환된 식의 평균과 분산은 모두 계산할 수 있기 때문에 정규분포의 PDF 에 대입하여 식을 정리할 수 있다.

$$N(x;\ \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$q(x_t \mid x_{t-1}) = N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t * I)$$
$$q(x_t \mid x_0) = N(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t) * I)$$

- $\alpha_t = 1 - \beta_t$
- $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$

$$\boldsymbol{q(x_{t-1} \mid x_t, x_0) = q(x_t \mid x_{t-1}) \frac{q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)}}$$

$$q(x_t|x_{t-1}) = \frac{1}{\sqrt{2\pi\beta_t}} \exp\left(-\frac{(x_t - \sqrt{1-\beta_t}x_{t-1})^2}{2\beta_t}\right)$$

$$q(x_t|x_0) = \frac{1}{\sqrt{2\pi(1-\bar{\alpha}_t)}} \exp\left(-\frac{(x_t - \sqrt{\bar{\alpha}_t}x_0)^2}{2(1-\bar{\alpha}_t)}\right)$$

$$q(x_{t-1}|x_0) = \frac{1}{\sqrt{2\pi(1-\bar{\alpha}_{t-1})}} \exp\left(-\frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0)^2}{2(1-\bar{\alpha}_{t-1})}\right)$$

식을 정리하면 PDF 공식에 따라 다음과 같이 $q(x_{t-1}|x_t,x_0)$의 평균(a)과 분산(b)을 알 수 있다.

$$\approx \frac{1}{\sqrt{2\pi\beta_t\left(\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\right)}} \exp\left(-\frac{1}{2\beta_t\left(\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\right)}\left[x_{t-1} - \left(\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_{t-1}}x_t\right)\right]^2\right)$$

b　　a

지금까지의 내용을 정리하면 forward 분포<$q(x_{t-1}|x_t, x_0)$>는 다음과 같이 x_0의 식으로 표현할 수 있다.

- $q(X_{t-1} \mid X_t) \rightarrow q(X_{t-1} \mid X_t, X_0)$
  - $N\left(X_{t-1};\ \tilde{\mu}(X_t, X_0), \tilde{\Sigma}(X_t, X_0)\right)$
  - $N\left(X_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_{t-1}}x_t\ ,\ \beta_t\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\right)$

forward 의 분포를 알았으니 이제 KLD 를 계산해보겠다. 두 분포가 정규분포일 때 KLD 의 공식은 다음과 같다.

$$D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \parallel \mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)) = \frac{1}{2}\left[\log\frac{|\boldsymbol{\Sigma}_y|}{|\boldsymbol{\Sigma}_x|} - d + \mathrm{tr}(\boldsymbol{\Sigma}_y^{-1}\boldsymbol{\Sigma}_x) + (\boldsymbol{\mu}_y - \boldsymbol{\mu}_x)^T\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{\mu}_y - \boldsymbol{\mu}_x)\right]$$

위 수식대로 식을 정리해보면 $q(x_{t-1}|x_t,x_0)$와 $p(x_{t-1}|x_t)$의 KLD 는 다음처럼 변경된다.

$$\arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))$$
$$= \arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q(t)) \parallel \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q(t)))$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\|_2^2\right]$$

결국 아래의 mu_q의 값을 예측하는 네트워크 mu_{theta}를 학습하면 된다.

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t}$$

그런데 여기서 저자는 한 단계 더 아이디어를 생각해냈다.

mu 를 한 번에 예측할 필요가 없이 x_0 를 예측해서 mu 를 계산할 수도 있다는 것이다. 다음과 같이 말이다.

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{1-\bar{\alpha}_t}$$

X_0 로 표현된 mu_q 와 mu_{theta} 를 정리된 KLD 식에 적용해보니 결국 x_0 에 대한 L2 loss 가 나왔다.

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} \left[\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\|_2^2\right]$$

근데 여기서 한 발 더 나아갈 수도 있다.

X_0 는 epsilon, X_t, a 의 조합으로 다음과 같이 표현할 수 있다.

$$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}}$$

그럼 mu_q 과 mu_theta 를 epsilon 으로도 표현할 수도 있다.

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0$$

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

그렇다면 같은 이치로 epsilon 을 예측하면 되는 것이 아닐까? 하고 정리된 KLD 식에 적용해보니 아래와 같은 수식이 나왔다고 한다.

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t} \left[\|\boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2\right]$$

그래서 결론은 mu($q(x_{t-1}|x_t, x_0)$의 평균)를 예측할 수도 있고, x_0 를 예측할 수도 있고, epsilon 을 예측할 수도 있다. 셋 다 forward<$q(x_{t-1}|x_t,x_0)$>와 reverse<$p(x_{t-1}|x_t)$>의 분포가 유사하게 만드는 것과 동치라는 의미이다.

지금까지의 내용을 정리하면 다음과 같다.

1. reverse 의 t-1 시점 이미지를 알기 위해 reverse 분포<$p(x_{t-1}|x_t)$>를 forward 분포<$q(x_{t-1}|x_t, x_0)$>에 근사하려고 한다. 이것은 KLD 로 학습이 가능하다.
2. forward 분포의 평균과 분산은 베이지안 룰로 계산할 수 있다.
3. 계산된 평균 수식에 위치한 x_0 를 epsilon 으로 다시 표현할 수 있다.
4. forward 와 reverse 분포의 KLD 는 forward 평균에 대한 L2 loss 로 정리된다.

5. L2 loss 에 epsilon 으로 정리된 평균 식(mu_q와 mu_{theta})을 대입하니 epsilon 을 예측하는 꼴로 변경된다.

- $q(X_{t-1} \mid X_t) \rightarrow q(X_{t-1} \mid X_t, X_0)$
  - $N\left(X_{t-1};\ \tilde{\mu}(X_t, X_0), \tilde{\Sigma}(X_t, X_0)\right)$
  - $N\left(X_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_{t-1}} x_t\ , \beta_t \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\right)$
  - $N\left(X_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon\right), \tilde{\beta}_t\right)$
  - $N\left(X_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t)\right), \tilde{\beta}_t\right)$

$$\arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) = \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \left[\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\|_2^2\right]$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t} \left[\|\boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2\right]$$

저자들은 세 가지 방법들 중 epsilon 을 이용하는 방법을 채택했다고 한다. 그 때가 가장 성능이 높다고 논문에 나와 있었다.

그래서 다시 수식을 써보면 아래와 같다.

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t} \left[\|\boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2\right]$$

마지막으로 저자들은 L2 distance 앞의 상수를 simple 하게 1 로 간주하여 다음과 같이 loss 를 정리했다고 한다.

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \epsilon}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right]$$

다음은 Training 알고리즘에 대한 설명이다.

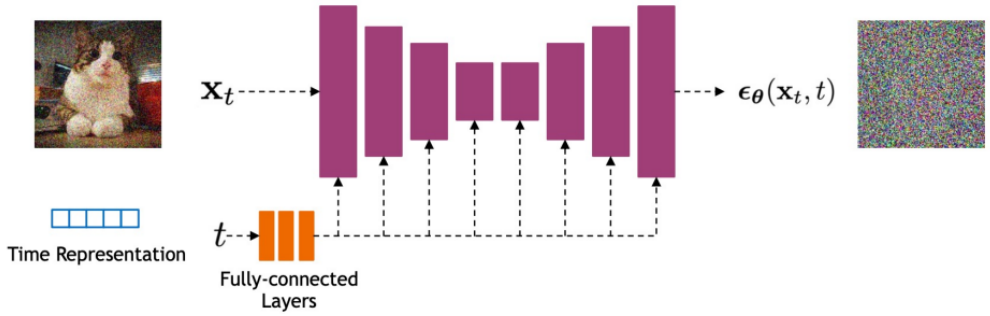
X_0 로부터 Uniform 한 t 를 추출하고, 표준 가우시안 분포에서 노이즈(epsilon)를 추출한다. 이후 노이즈가 추가된 t step 에서의 이미지를 만들고, 이 이미지를 만들 때 사용한 t step 에서의 노이즈가 무엇인지 예측하기 위해 실제 추가된 노이즈(e)와의 L2 loss 로 학습한다.

(모든 step 에서 노이즈 예측하지 않고, t step 에서만 예측하는 구조)

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

네트워크 구조는 다음과 같이 생겼다. Time step 을 컨디션 더하는 것처럼 모델에 더해주는 방식이다.



다음은 샘플링에 대한 설명이다. 샘플링 과정에서는 모든 t step 에 대하여 reverse process 를 진행한다. t step 이미지(X_t)와 t step 에서 예측된 노이즈(epsilon)를 알고 있을 때, X_{t-1} 을 계산할 수 있다. 그 이유는 모델이 노이즈를 예측하면 forward 분포<$q(x_{t-1}|x_t,x_0)$>의 평균을 (근사해서) 계산할 수 있고, 이렇게 계산된 값을 reverse 분포<$p(x_{t-1}|x_t)$>의 평균으로 간주하기 때문에 X_{t-1} 는 reparameterization trick 으로 계산할 수 있다.

만일 노이즈를 예측하지 않고 x_0 를 예측하였다면, 평균에 대한 수식이 바뀔 테니 그 수식에만 잘 적용하면 된다. 그리고 표준편차 같은 경우, 분산이 B_t 나 마찬가지이므로 z 에 B_t 를 곱해도 된다고 한다.

Algorithm 2 Sampling

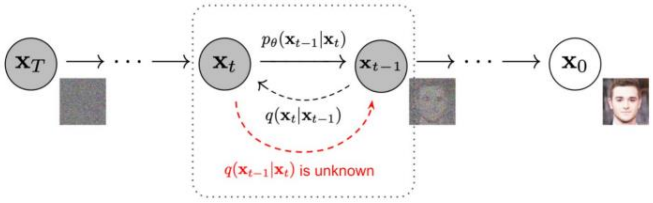
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

$$N\left(X_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t)\right), \tilde{\beta}_t\right) \quad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

마지막으로 DDPM 을 정리하면 다음 그림과 같이 세 식으로 나타낼 수 있다.

1. Forward 과정은 alpha 와 noise 를 이용한 선형보간법으로 계산한다. (=)Reparameterization trick
2. Loss 는 x_0 가 x_t 가 되기 위해 사용된 noise 를 예측하는 방식이다.
3. Reverse 과정은 모든 t step 에 대해 noise 를 예측하고 forward 분포의 평균<$q(x_{t-1}|x_t, x_0)$>을 계산한 다음, reparameteriazation trick 으로 t-1 시점의 이미지를 제작한다.



Denoising Diffusion Probabilistic Models

- $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon,\ \epsilon \sim \mathcal{N}(0, \mathbf{I})$ (Forward)
  - $\beta_1 = 10^{-4}, \beta_T = 0.02$
  - $\alpha_t := 1 - \beta_t,\ \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$
- $\epsilon - \epsilon_\theta(\mathbf{x}_t) = \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon)$ (Loss)
  - ϵ_θ = prediction network
- $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) + \tilde{\beta}_t\epsilon,\ \epsilon \sim \mathcal{N}(0, \mathbf{I})$ (Reverse)
  - $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$
    - $\tilde{\beta}_t = \beta_t$로 해도 성능차이 없음

DDIM 의 간단한 이해

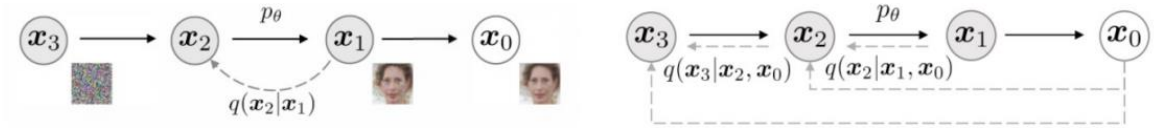


Figure 1: Graphical models for diffusion (left) and non-Markovian (right) inference models.

Denoising Diffusion Implicit Models

DDIM 은 DDPM 에서 Reverse process 를 T 번만큼 수행하는 이유가 마르코프 체인의 성질 때문이라고 설명하고, non-Markov 하게 forward diffusion process 를 진행한다.
그래서 DDPM 과는 달리 x_3 에 대한 분포를 계산할 때 x_2 만 필요한 것이 아니라, x_0 도 필요하다고 주장한다. 정리하면 DDIM 은 X_t 가 바로 이전 step 값인 X_{t-1} 과 X_0 에 의해 결정되는 Non-markovian chain 개념을 이용한다.

따라서 forward 의 분포<$q(x_{t-1}|x_t)$>가 DDPM 에서는 X_0 를 이용하지 않고도 표현을 할 수 있었는데, DDIM<$q_\sigma(x_{t-1}|x_t)$>에서는 항상 X_0 를 이용해서 표현하는 방식을 이용한다고 한다.

- $q(X_{t-1} \mid X_t) \rightarrow q(X_{t-1} \mid X_t, X_0)$
  - $N\left(X_{t-1};\ \tilde{\mu}(X_t, X_0), \tilde{\Sigma}(X_t, X_0)\right)$
  - $N\left(X_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon\right), \tilde{\beta}_t\right)$

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\boldsymbol{I}\right)$$

수식을 비교하면 DDPM(q)는 평균을 x_t에 대한 식으로 표현했지만,
DDIM(q_σ)는 평균을 X_t와 X_0으로 표현하였다.

참고로 위의 수식은 DDPM 에서 [x_0와 epsilon 을 이용해서 x_{t-1}를 표현하는 수식]을 아래와 같이 전개해서 구했다고 한다.

$$
\begin{aligned}
\mathbf{x}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_t + \sigma_t\boldsymbol{\epsilon} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} + \sigma_t\boldsymbol{\epsilon} \\
q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I})
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} && \text{;where } \boldsymbol{\epsilon}_{t-1}, \boldsymbol{\epsilon}_{t-2}, \cdots \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2} && \text{;where } \bar{\boldsymbol{\epsilon}}_{t-2} \text{ merges two Gaussians (*).} \\
&= \dots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \\
q(\mathbf{x}_t|\mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})
\end{aligned}
$$

(*) Recall that when we merge two Gaussians with different variance, $\mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$ and $\mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I})$, the new distribution is $\mathcal{N}(\mathbf{0}, (\sigma_1^2 + \sigma_2^2)\mathbf{I})$. Here the merged standard deviation is $\sqrt{(1-\alpha_t) + \alpha_t(1-\alpha_{t-1})} = \sqrt{1-\alpha_t\alpha_{t-1}}$.

이 때, 위 그림의 두 번째 식에서 x_0와 epsilon 을 (epsilon 를 예측하는) Neural Network 를 통해 계산하여 바꿔 쓴다면 다음과 같이 표현할 수도 있다. (epsilon 의 아래 첨자는 무시하고 생각하자)

$$
\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}}
$$

$$
\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}}\underbrace{\left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}{\sqrt{\alpha_t}}\right)}_{\text{“ predicted } \boldsymbol{x}_0\text{”}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}_{\text{“direction pointing to } \boldsymbol{x}_t\text{”}} + \underbrace{\sigma_t\epsilon_t}_{\text{random noise}}
$$

따라서 기존 DDPM 과는 달리 network 에서 [예측된 epsilon 를 통해 구한 x_0, x_t]를 모두 이용해서 Reverse process 를 진행한다고 생각하면 된다. 또한 논문에서 학습 과정은 DDPM 과 같다고 나와 있으므로, DDIM 을 정리하면 아래와 같이 표현할 수 있다.

- DDIM α = DDPM $\bar{\alpha}$
- $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon$ (Forward)
- $\epsilon - \epsilon_\theta(\mathbf{x}_t) = \epsilon - \epsilon_\theta(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon)$ (Loss)
  - ϵ_θ = prediction network
- $\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\underbrace{\left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t)}{\sqrt{\alpha_t}}\right)}_{\text{predicted } \mathbf{x}_0 = f_\theta(\mathbf{x}_t)} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta(\mathbf{x}_t)}_{\text{direction pointing to } \mathbf{x}_t} + \underbrace{\sigma_t\epsilon}_{\text{noise}}$ (Reverse)
  - deterministic when $\sigma_t = 0$ → consistency

추정된 x_0까지 이용하여 X_{t-1}를 계산한다면, 처음의 시작점(x_0)을 알기 때문에(힌트가 주어지므로) 몇 step 정도는 뛰어 넘고 다음 이미지를 만들 수 있다고 논문에서 주장한다. (단, sigma 를 0 으로 두어야 한다고 논문에서 제시됨)