Reading ~~Argmax Flows and~~ *Multinomial Diffusion: Learning Categorical Distributions*

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, Max Welling (*presented by Joseph Le Roux*)

RCLN reading group 30/09/24

Outline

Introduction

A mecha robot playing the guitar in a forest, low quality, 3d, photorealistic

Diffusion Models are known to be good at generating *realistic* images.

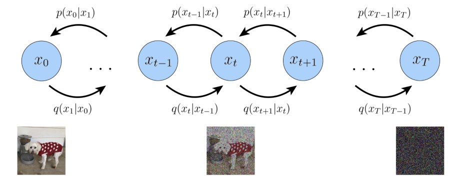
Can they be used to generate texts ?

Several papers, different models We focus on the first one in this talk.

Diffusion Models (2)

Two reciprocal processes: forward and backward

- diffusion distribution q (*forward*) generates noise from data
- generation generates data as denoising via distribution p (*backward*)

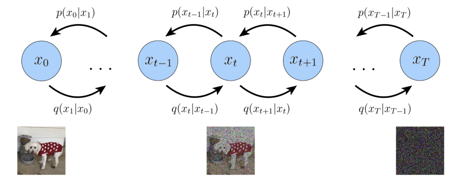


- q is fixed, we want to learn p

Diffusion Models (2)

Two reciprocal processes: forward and backward

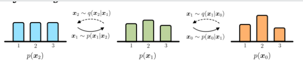
- diffusion distribution q (*forward*) generates noise from data
- generation generates data as denoising via distribution p (*backward*)



- q is fixed, we want to learn p

For discrete distributions

This paper discusses how to model diffusion for *multinomial* distributions (MD)



(b) Multinomial Diffusion: Each step $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ denoises the signal starting from a uniform categorical base distribution which gives the model $p(\boldsymbol{x}_0)$.

Model

Definition

We want to generate data from a target multinomial distribution of K classes.

Data

We denote:

- $\boldsymbol{x}_0$, a piece of data generated by the target MD;
- $\boldsymbol{x}_t$, a piece of data generated by a noisy version of target MD after t forward steps.

$\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T$ are all one-hot vectors of length K.

Define a diffusion model

We need 2 conditional probabilities:

- forward $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$
- backward $p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})$

Forward diffusion process

$q(x_t|x_{t-1})$

- many possibilities, must be easy to sample from;
- in this paper:
  1. flip a (biased) coin;
  2. if head then do not chang x_{t-1}, else (tail), choose a category at random (uniformly)

This amounts to:

$$q(x_t|x_{t-1} = \boldsymbol{\delta}_k) = \begin{cases} (1-\beta_t) + \frac{\beta_t}{K} & \text{if } x_t = \boldsymbol{\delta}_k \\ \frac{\beta_t}{K} & \text{otherwise.} \end{cases}$$

where β_t is a hyper-parameter

But we will need more

- $q(x_t|x_0)$ but be easily computable (apply t forward steps in a row)
- the *posterior* $q(x_{t-1}|x_t, x_0)$ must also be easy to compute

Combining steps of forward process

Combine 2 steps

$$q(x_{t+1}|x_{t-1} = \boldsymbol{\delta}_k) = \sum_{x_t} q(x_t|x_{t-1} = \boldsymbol{\delta}_k) q(x_{t+1}|x_t, x_{t-1} = \boldsymbol{\delta}_k)$$

Combinining t steps from the beginning

Combining steps of forward process

Combine 2 steps

$$
\begin{aligned}
q(x_{t+1}|x_{t-1} = \boldsymbol{\delta}_k) &= \sum_{x_t} q(x_t|x_{t-1} = \boldsymbol{\delta}_k) q(x_{t+1}|x_t, x_{t-1} = \boldsymbol{\delta}_k) \\
&= \sum_{x_t} q(x_t|x_{t-1} = \boldsymbol{\delta}_k) q(x_{t+1}|x_t)
\end{aligned}
$$

Combinining t steps from the beginning

Combining steps of forward process

Combine 2 steps

$$
\begin{aligned}
q(x_{t+1}|x_{t-1} = \boldsymbol{\delta}_k) &= \sum_{x_t} q(x_t|x_{t-1} = \boldsymbol{\delta}_k) q(x_{t+1}|x_t, x_{t-1} = \boldsymbol{\delta}_k) \\
&= \sum_{x_t} q(x_t|x_{t-1} = \boldsymbol{\delta}_k) q(x_{t+1}|x_t) \\
&= q(x_t = \boldsymbol{\delta}_k|x_{t-1} = \boldsymbol{\delta}_k) q(x_{t+1}|x_t = \boldsymbol{\delta}_k) + \sum_{x_t \neq \boldsymbol{\delta}_k} q(x_t|x_{t-1} = \boldsymbol{\delta}_k) q(x_{t+1}|x_t)
\end{aligned}
$$

Combinining t steps from the beginning

Combining steps of forward process

Combine 2 steps

$$
\begin{aligned}
q(\mathbf{x}_{t+1}|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k) &= \sum_{\mathbf{x}_t} q(\mathbf{x}_t|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{x}_{t-1}=\boldsymbol{\delta}_k) \\
&= \sum_{\mathbf{x}_t} q(\mathbf{x}_t|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t) \\
&= q(\mathbf{x}_t=\boldsymbol{\delta}_k|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t=\boldsymbol{\delta}_k) + \sum_{\mathbf{x}_t\neq\boldsymbol{\delta}_k} q(\mathbf{x}_t|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t) \\
&= \begin{cases} ((1-\beta_t)+\frac{\beta_t}{K})((1-\beta_{t+1})+\frac{\beta_{t+1}}{K})+(K-1)\frac{\beta_t}{K}\frac{\beta_{t+1}}{K} & \text{if } \mathbf{x}_{t+1}=\boldsymbol{\delta}_k \\ \frac{1-\text{above}}{K-1} & \text{otherwise} \end{cases}
\end{aligned}
$$

Combinining t steps from the beginning

Combining steps of forward process

Combine 2 steps

$$
\begin{aligned}
q(\mathbf{x}_{t+1}|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k) &= \sum_{\mathbf{x}_t} q(\mathbf{x}_t|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{x}_{t-1}=\boldsymbol{\delta}_k) \\
&= \sum_{\mathbf{x}_t} q(\mathbf{x}_t|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t) \\
&= q(\mathbf{x}_t=\boldsymbol{\delta}_k|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t=\boldsymbol{\delta}_k) + \sum_{\mathbf{x}_t\neq\boldsymbol{\delta}_k} q(\mathbf{x}_t|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t) \\
&= \begin{cases} ((1-\beta_t)+\frac{\beta_t}{K})((1-\beta_{t+1})+\frac{\beta_{t+1}}{K}) + (K-1)\frac{\beta_t}{K}\frac{\beta_{t+1}}{K} & \text{if } \mathbf{x}_{t+1}=\boldsymbol{\delta}_k \\ \frac{1-\text{above}}{K-1} & \text{otherwise} \end{cases} \\
&= \begin{cases} (1-\beta_t)(1-\beta_{t+1}) + \frac{1-(1-\beta_t)(1-\beta_{t+1})}{K} & \text{if } \mathbf{x}_{t+1}=\boldsymbol{\delta}_k \\ \frac{1-(1-\beta_t)(1-\beta_{t+1})}{K} & \text{otherwise.} \end{cases}
\end{aligned}
$$

Combinining t steps from the beginning

Combining steps of forward process

Combine 2 steps

$$
\begin{aligned}
q(\mathbf{x}_{t+1}|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k) &= \sum_{\mathbf{x}_t} q(\mathbf{x}_t|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t,\mathbf{x}_{t-1}=\boldsymbol{\delta}_k) \\
&= \sum_{\mathbf{x}_t} q(\mathbf{x}_t|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t) \\
&= q(\mathbf{x}_t=\boldsymbol{\delta}_k|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t=\boldsymbol{\delta}_k) + \sum_{\mathbf{x}_t\neq\boldsymbol{\delta}_k} q(\mathbf{x}_t|\mathbf{x}_{t-1}=\boldsymbol{\delta}_k)q(\mathbf{x}_{t+1}|\mathbf{x}_t) \\
&= \begin{cases} ((1-\beta_t)+\frac{\beta_t}{K})((1-\beta_{t+1})+\frac{\beta_{t+1}}{K})+(K-1)\frac{\beta_t}{K}\frac{\beta_{t+1}}{K} & \text{if } \mathbf{x}_{t+1}=\boldsymbol{\delta}_k \\ \frac{1-\text{above}}{K-1} & \text{otherwise} \end{cases} \\
&= \begin{cases} (1-\beta_t)(1-\beta_{t+1}) + \frac{1-(1-\beta_t)(1-\beta_{t+1})}{K} & \text{if } \mathbf{x}_{t+1}=\boldsymbol{\delta}_k \\ \frac{1-(1-\beta_t)(1-\beta_{t+1})}{K} & \text{otherwise.} \end{cases}
\end{aligned}
$$

Combinining t steps from the beginning

$$
q(\mathbf{x}_t|\mathbf{x}_0=\boldsymbol{\delta}_k) = \begin{cases} \bar{\alpha}_t + \frac{1-\bar{\alpha}_t}{K} & \text{if} \mathbf{x}_t=\boldsymbol{\delta}_k \\ \frac{1-\bar{\alpha}_t}{K} & \text{otherwise.} \end{cases}
$$

{with $\alpha_t = \prod_{i=0}^{t}(1-\beta_i)$ and $\bar{\alpha}_t = 1-\alpha_t$}

Computing the posterior

The posterior will be needed in the loss function

Derivation of posterior

$$
\begin{aligned}
q(x_{t-1}|x_t, x_0) &= \frac{q(x_{t-1}, x_t|x_0)}{q(x_t|x_0)} \\
&= \frac{q(x_t|x_{t-1}, x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)} \\
&= \frac{q(x_t|x_{t-1})q(x_{t-1}|x_0)}{q(x_t|x_0)}
\end{aligned}
$$

Computing the posterior

The posterior will be needed in the loss function

Derivation of posterior

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

For concrete data values we have:

$$q(x_{t-1} = \boldsymbol{\delta}_p|x_t = \boldsymbol{\delta}_c, x_0 = \boldsymbol{\delta}_k) = \frac{q(x_t = \boldsymbol{\delta}_c|x_{t-1} = \boldsymbol{\delta}_p)q(x_{t-1} = \boldsymbol{\delta}_p|x_0 = \boldsymbol{\delta}_k)}{q(x_t = \boldsymbol{\delta}_c|x_0 = \boldsymbol{\delta}_k)}$$
$$= \frac{q(x_t = \boldsymbol{\delta}_c|x_{t-1} = \boldsymbol{\delta}_p)q(x_{t-1} = \boldsymbol{\delta}_p|x_0 = \boldsymbol{\delta}_k)}{\sum_{\boldsymbol{\delta}_{p'}} q(x_t = \boldsymbol{\delta}_c, x_{t-1} = \boldsymbol{\delta}_{p'}|x_0 = \boldsymbol{\delta}_k)}$$
$$= \frac{q(x_t = \boldsymbol{\delta}_c|x_{t-1} = \boldsymbol{\delta}_p)q(x_{t-1} = \boldsymbol{\delta}_p|x_0 = \boldsymbol{\delta}_k)}{\sum_{\boldsymbol{\delta}_{p'}} q(x_t = \boldsymbol{\delta}_c|x_{t-1} = \boldsymbol{\delta}_{p'})q(x_{t-1} = \boldsymbol{\delta}_{p'}|x_0 = \boldsymbol{\delta}_k)}$$
$$= \frac{\boldsymbol{\theta}(\delta_k, \delta_c)_p}{\sum_{p'=1}^{K} \boldsymbol{\theta}(\delta_k, \delta_c)_{p'}}$$

Take home message: we can precompute all posteriors and store them in tables.

Backward Process $p(x_{t-1}|x_t)$ as Denoising

- The distribution that we want to learn and implement via a neural network
- Actually, T distributions, with different behaviours: too difficult
- So we rewrite the backward process and model only a part of it

Denoising with posterior

To denoise from step t to step $t-1$:

1. complete denoising: predict clean from noisy (ie perform t backward steps)
2. from predicted data use the posterior to perform $(t-1)$ forward steps.

$$\begin{aligned} p(x_{t-1}|x_t) &= \sum_{x_0} p(x_{t-1}, x_0|x_t) \\ &= \sum_{x_0} p(x_0|x_t)p(x_{t-1}|x_0, x_t) \\ &= \sum_{x_0} p(x_0|x_t)q(x_{t-1}|x_0, x_t) \\ &= q(x_{t-1}|\hat{x_0}, x_t) \text{ with } \hat{x_0} = \mu(x_t, t) \end{aligned}$$

Remarks

1. $\hat{x_0}$ is ≥ 0, sums to 1, but not one-hot.
2. $p(x_0|x_1) = q(x_0|\hat{x_0}, x_1)$ is simply $\hat{x_0} = \mu(x_1, 1)$ seen as a distribution.

Learning Diffusion Models

Learning Problem (1)

Maximize the log-likelihood with latent diffusion

$$\log p(x_0) = \log \sum_{x_1,\ldots,x_T} p(x_0, x_1, \ldots, x_T)$$

Learning Problem (1)

Maximize the log-likelihood with latent diffusion

$$
\begin{aligned}
\log p(x_0) &= \log \sum_{x_1,\ldots,x_T} p(x_0, x_1, \ldots, x_T) \\
&= \log \sum_{x_1,\ldots,x_T} \frac{q(x_1, \ldots, x_T | x_0)}{q(x_1, \ldots, x_T | x_0)} p(x_0, x_1, \ldots, x_T)
\end{aligned}
$$

Learning Problem (1)

Maximize the log-likelihood with latent diffusion

$$
\begin{aligned}
\log p(x_0) &= \log \sum_{x_1,\dots,x_T} p(x_0, x_1, \dots, x_T) \\
&= \log \sum_{x_1,\dots,x_T} \frac{q(x_1,\dots,x_T|x_0)}{q(x_1,\dots,x_T|x_0)} p(x_0, x_1, \dots, x_T) \\
&= \log \mathbb{E}_{x_1,\dots,x_T \sim q} [\frac{p(x_0, x_1, \dots, x_T)}{q(x_1,\dots,x_T|x_0)}]
\end{aligned}
$$

Learning Problem (1)

Maximize the log-likelihood with latent diffusion

$$
\begin{aligned}
\log p(x_0) &= \log \sum_{x_1,\ldots,x_T} p(x_0, x_1, \ldots, x_T) \\
&= \log \sum_{x_1,\ldots,x_T} \frac{q(x_1,\ldots,x_T|x_0)}{q(x_1,\ldots,x_T|x_0)} p(x_0, x_1, \ldots, x_T) \\
&= \log \mathbb{E}_{x_1,\ldots,x_T \sim q}[\frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&\geq \mathbb{E}_{x_1,\ldots,x_T \sim q}[\log \frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}]
\end{aligned}
$$

Learning Problem (1)

Maximize the log-likelihood with latent diffusion

$$
\begin{aligned}
\log p(x_0) &= \log \sum_{x_1,\ldots,x_T} p(x_0, x_1, \ldots, x_T) \\
&= \log \sum_{x_1,\ldots,x_T} \frac{q(x_1,\ldots,x_T|x_0)}{q(x_1,\ldots,x_T|x_0)} p(x_0, x_1, \ldots, x_T) \\
&= \log \mathbb{E}_{x_1,\ldots,x_T \sim q}\left[\frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}\right] \\
&\geq \mathbb{E}_{x_1,\ldots,x_T \sim q}\left[\log \frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}\right] \\
&= \mathbb{E}_{x_1,\ldots,x_T \sim q}\left[\log \frac{p(x_T)p(x_0, x_1, \ldots|x_T)}{q(x_1,\ldots,x_T|x_0)}\right]
\end{aligned}
$$

Learning Problem (1)

Maximize the log-likelihood with latent diffusion

$$
\begin{aligned}
\log p(x_0) &= \log \sum_{x_1,\ldots,x_T} p(x_0, x_1, \ldots, x_T) \\
&= \log \sum_{x_1,\ldots,x_T} \frac{q(x_1,\ldots,x_T|x_0)}{q(x_1,\ldots,x_T|x_0)} p(x_0, x_1, \ldots, x_T) \\
&= \log \mathbb{E}_{x_1,\ldots,x_T\sim q}[\frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&\geq \mathbb{E}_{x_1,\ldots,x_T\sim q}[\log \frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&= \mathbb{E}_{x_1,\ldots,x_T\sim q}[\log \frac{p(x_T)p(x_0, x_1, \ldots|x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&= \mathbb{E}_{x_1,\ldots,x_T\sim q}[\log \frac{p(x_T)\prod_{t=1}^{T} p(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1})}]
\end{aligned}
$$

Learning Problem (1)

Maximize the log-likelihood with latent diffusion

$$
\begin{aligned}
\log p(x_0) &= \log \sum_{x_1,\ldots,x_T} p(x_0, x_1, \ldots, x_T) \\
&= \log \sum_{x_1,\ldots,x_T} \frac{q(x_1,\ldots,x_T|x_0)}{q(x_1,\ldots,x_T|x_0)} p(x_0, x_1, \ldots, x_T) \\
&= \log \mathbb{E}_{x_1,\ldots,x_T\sim q}[\frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&\geq \mathbb{E}_{x_1,\ldots,x_T\sim q}[\log \frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&= \mathbb{E}_{x_1,\ldots,x_T\sim q}[\log \frac{p(x_T)p(x_0, x_1, \ldots|x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&= \mathbb{E}_{x_1,\ldots,x_T\sim q}[\log \frac{p(x_T)\prod_{t=1}^{T} p(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1})}] \\
&= \mathbb{E}_{x_1,\ldots,x_T\sim q}[\log p(x_T) + \sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}]
\end{aligned}
$$

Learning Problem (1)

Maximize the log-likelihood with latent diffusion

$$
\begin{aligned}
\log p(x_0) &= \log \sum_{x_1,\ldots,x_T} p(x_0, x_1, \ldots, x_T) \\
&= \log \sum_{x_1,\ldots,x_T} \frac{q(x_1,\ldots,x_T|x_0)}{q(x_1,\ldots,x_T|x_0)} p(x_0, x_1, \ldots, x_T) \\
&= \log \mathbb{E}_{x_1,\ldots,x_T \sim q}[\frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&\geq \mathbb{E}_{x_1,\ldots,x_T \sim q}[\log \frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&= \mathbb{E}_{x_1,\ldots,x_T \sim q}[\log \frac{p(x_T)p(x_0, x_1, \ldots|x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&= \mathbb{E}_{x_1,\ldots,x_T \sim q}[\log \frac{p(x_T)\prod_{t=1}^{T} p(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1})}] \\
&= \mathbb{E}_{x_1,\ldots,x_T \sim q}[\log p(x_T) + \sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}]
\end{aligned}
$$

- {Maximize last line, a lower bound of the log-likelihood, as a *surrogate* loss.}

Learning Problem (1)

Maximize the log-likelihood with latent diffusion

$$
\begin{aligned}
\log p(x_0) &= \log \sum_{x_1,\ldots,x_T} p(x_0, x_1, \ldots, x_T) \\
&= \log \sum_{x_1,\ldots,x_T} \frac{q(x_1,\ldots,x_T|x_0)}{q(x_1,\ldots,x_T|x_0)} p(x_0, x_1, \ldots, x_T) \\
&= \log \mathbb{E}_{x_1,\ldots,x_T \sim q}[\frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&\geq \mathbb{E}_{x_1,\ldots,x_T \sim q}[\log \frac{p(x_0, x_1, \ldots, x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&= \mathbb{E}_{x_1,\ldots,x_T \sim q}[\log \frac{p(x_T)p(x_0, x_1, \ldots |x_T)}{q(x_1,\ldots,x_T|x_0)}] \\
&= \mathbb{E}_{x_1,\ldots,x_T \sim q}[\log \frac{p(x_T)\prod_{t=1}^{T} p(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1})}] \\
&= \mathbb{E}_{x_1,\ldots,x_T \sim q}[\log p(x_T) + \sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}]
\end{aligned}
$$

- {Maximize last line, a lower bound of the log-likelihood, as a *surrogate* loss.}
- {... but because of sampling, this has high variance, we need more maths!}

Learning Problem (2)

Forget constant terms

$$\mathbb{E}_q[\log p(x_T) + \sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] = \mathbb{E}_q[\log p(x_T)] + \mathbb{E}_q[\sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}]$$

Use the special definition of $p(x_0|x_1)$

Learning Problem (2)

Forget constant terms

$$\mathbb{E}_q[\log p(x_T) + \sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] = \mathbb{E}_q[\log p(x_T)] + \mathbb{E}_q[\sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}]$$

$$= C + \mathbb{E}_q[\sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}]$$

Use the special definition of $p(x_0|x_1)$

Learning Problem (2)

Forget constant terms

$$\mathbb{E}_q[\log p(x_T) + \sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] = \mathbb{E}_q[\log p(x_T)] + \mathbb{E}_q[\sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}]$$

$$= C + \mathbb{E}_q[\sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}]$$

Use the special definition of $p(x_0|x_1)$

$$\mathbb{E}_q[\sum_{t=1}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] = \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] + \mathbb{E}_q[\log \frac{p(x_0|x_1)}{q(x_1|x_0)}]$$

$$= \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] + \mathbb{E}_q[\log \frac{\mu(x_1, 1)}{q(x_1|x_0)}]$$

$$= \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] + \mathbb{E}_q[\log \mu(x_1, 1)] - C$$

Learning Problem (3)

Use the posterior

$$\mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] = \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}]$$

Use Kullback-Liebler divergence

Learning Problem (3)

Use the posterior

$$\mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] = \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}]$$

$$= \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}] + \mathbb{E}_q[\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}]$$

Use Kullback-Liebler divergence

Learning Problem (3)

Use the posterior

$$\mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}] = \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}]$$

$$= \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}] + \mathbb{E}_q[\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}]$$

$$= \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}] + C$$

Use Kullback-Liebler divergence

Learning Problem (3)

Use the posterior

$$\mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] = \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}]$$

$$= \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}] + \mathbb{E}_q[\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}]$$

$$= \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}] + C$$

Use Kullback-Liebler divergence

$$\mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}] = \sum_{t=2}^{T} \mathbb{E}_q[\log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}]$$

Learning Problem (3)

Use the posterior

$$\mathbb{E}_q[\sum_{t=2}^{T}\log\frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] = \mathbb{E}_q[\sum_{t=2}^{T}\log\frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}]$$

$$= \mathbb{E}_q[\sum_{t=2}^{T}\log\frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}] + \mathbb{E}_q[\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}]$$

$$= \mathbb{E}_q[\sum_{t=2}^{T}\log\frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}] + C$$

Use Kullback-Liebler divergence

$$\mathbb{E}_q[\sum_{t=2}^{T}\log\frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}] = \sum_{t=2}^{T}\mathbb{E}_q[\log\frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}]$$

$$= \sum_{t=2}^{T}\mathbb{E}_q[-KL(q(x_{t-1}|x_t,x_0)||p(x_{t-1}|x_t))]$$

Learning Problem (3)

Use the posterior

$$\mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_t|x_{t-1})}] = \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}]$$

$$= \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}] + \mathbb{E}_q[\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}]$$

$$= \mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}] + C$$

Use Kullback-Liebler divergence

$$\mathbb{E}_q[\sum_{t=2}^{T} \log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}] = \sum_{t=2}^{T} \mathbb{E}_q[\log \frac{p(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}]$$

$$= \sum_{t=2}^{T} \mathbb{E}_q[-KL(q(x_{t-1}|x_t, x_0)||p(x_{t-1}|x_t))]$$

$$= \sum_{t=2}^{T} \mathbb{E}_q[-KL(q(x_{t-1}|x_t, x_0)||q(x_{t-1}|x_t, \hat{x_0}))]$$

Learning Problem (4)

From our definitions of posteriors

$$-KL\big(q(x_{t-1}|x_t, x_0)||q(x_{t-1}|x_t, \hat{x_0})\big) = -KL\Big(\frac{\boldsymbol{\theta}(x_0, x_t)[x_{t-1}]}{\sum \boldsymbol{\theta}(x_0, x_t)}||\frac{\boldsymbol{\theta}(\hat{x_0}, x_t)[x_{t-1}]}{\sum \boldsymbol{\theta}(\hat{x_0}, x_t)}\Big)$$

Learning Problem (4)

From our definitions of posteriors

$$-KL\big(q(x_{t-1}|x_t, x_0)||q(x_{t-1}|x_t, \hat{x_0})\big) = -KL\Big(\frac{\boldsymbol{\theta}(x_0, x_t)[x_{t-1}]}{\sum \boldsymbol{\theta}(x_0, x_t)}||\frac{\boldsymbol{\theta}(\hat{x_0}, x_t)[x_{t-1}]}{\sum \boldsymbol{\theta}(\hat{x_0}, x_t)}\Big)$$

Congratulations! You (and I) survived

We have our loss function defined as:

$$\mathcal{L}(x_0) = \mathbb{E}_q \log p(x_0|x_1) + \sum_{t=2}^{T} \mathbb{E}_q\Big[-KL\Big(\frac{\boldsymbol{\theta}(x_0, x_t)[x_{t-1}]}{\sum \boldsymbol{\theta}(x_0, x_t)}||\frac{\boldsymbol{\theta}(\hat{x_0}, x_t)[x_{t-1}]}{\sum \boldsymbol{\theta}(\hat{x_0}, x_t)}\Big)\Big]$$

In practice, do not optimize for every timestep:

- sample $1 \leq t \leq T$ at random
- diffuse x_0 for t timesteps (or better, sample from $q(x_t|x_0)$ directly)
- optimize KL for timestep t only
- move to the next example

Applications

Experiments on Language

Datasets

- `text8`: *data has First billion characters from wikipedia (clean data), can be used in word2vec, glove etc*
  - 27 categories (26 letters + space)
  - chunked in sequences of length 256
  - train/dev/test sizes: 90000000/5000000/5000000
- `enwik8`: *first 100,000,000 (100M) bytes of the English Wikipedia XML dump on Mar. 3, 2006 and is typically used to measure a model's ability to compress data*
  - 256 categories (bytes)
  - chunked in sequences of length 320
  - train/dev/test sizes: 90000000/5000000/5000000

Architecture

- 12 layer transformer (encoders only), 8 heads, layer size is 512
- 1000 diffusion steps for `text8`
- 4000 diffusion steps for `enwik8`

Results (1)

Compression metrics

Table 3: Comparison of different methods on `text8` and `enwik8`. Results are reported in negative log-likelihood with units bits per character (bpc) for `text8` and bits per raw byte (bpb) for `enwik8`.

Model type	Model	text8 (bpc)	enwik8 (bpb)
ARM	64 Layer Transformer (Al-Rfou et al., 2019)	1.13	1.06
	TransformerXL (Dai et al., 2019)	1.08	0.99
VAE	AF/AF* (AR) (Ziegler and Rush, 2019)	1.62	1.72
	IAF / SCF* (Ziegler and Rush, 2019)	1.88	2.03
	CategoricalNF (AR) (Lippe and Gavves, 2020)	1.45	-
Generative Flow	Argmax Flow, AR (ours)	1.39	1.42
	Argmax Coupling Flow (ours)	1.82	1.93
Diffusion	Multinomial Text Diffusion (ours)	1.72	1.75

⋆ Results obtained by running code from the official repository for the `text8` and `enwik8` datasets.

- worse than autoregressive models
- better than non-AR with continuous embeddings

Sampling

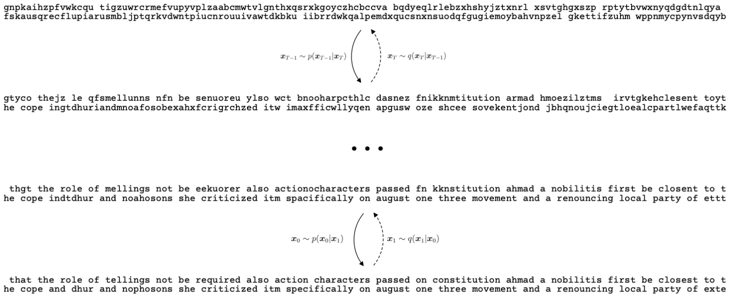


Figure 7: Intermediate steps of the generation chain of the Multinomial Text Diffusion model trained on `text8`.

Results (3)

Spell Checking

as a by-product, assume that input text is x_1 and predict x_0

```
mexico city the aztec stadium estadio azteca home of club america is on
e of the world s largest stadiums with capacity to seat approximately o
ne one zero zero zero zero fans mexico hosted the football world cup in
 one nine seven zero and one nine eight six
```

(a) Ground truth sequence from `text8`.

```
mexico citi the aztec stadium estadio azteca home of clup amerika is on
e of the world s largest stadioms with capakity to seat approsimately o
ne one zeto zero zero zero fans mexico hosted the footpall wolld cup in
 one nine zeven zero and one nyne eiggt six
```

(b) Corrupted sentence.

```
mexico city the aztec stadium estadio aztecs home of club america is on
e of the world s largest stadiums with capacity to seat approximately o
ne one zero zero zero zero fans mexico hosted the football world cup in
 one nine seven zero and one nine eight six
```

(c) Suggested, prediction by the model.

Figure 5: Spell checking with Multinomial Text Diffusion.

Conclusion

Conclusion

Summary

- First attempt to define a diffusion model for discrete multinomial data
  - data is discrete
  - time is discrete
- Language modelling performance worse than autoregressive
- prediction does not depend (directly) on the length of the generated sequence

The Future (of this paper)

- discrete diffusion and autoregressive models
- continuous time
- score-matching for complex discrete data (enery-basde models)