Diffusion Model Learning

Created	@2025年4月21日 19:05
Class	1216

The Learning of Diffusion Model

In this paper,I'll start to record my learning process about diffusion model. At the same time,it will analysis several classic papers in diffusion model. OK,Let's go!

1 Essential Foundational Knowledge

1. common function

LaTeX代码	含义
`=`	等于
`\neq`	不等于
`\approx`	约等于
`\sim`	相似于 / 分布符号
`\propto`	正比于
`\equiv`	恒等于
`\geq`	大于等于
`\leq`	小于等于
`\gg`	远大于
`\ll`	远小于
`\in`	属于（集合）
`\notin`	不属于（集合）
`\subset`	子集
`\subseteq`	包含于
`\supset`	超集
`\cup`	并集
`\cap`	交集
`\forall`	对所有（全称量词）
`\exists`	存在（存在量词）

2. Markov Chain：

A Markov chain is a mathematical system that undergoes transitions from one state to another.It is a random process, but it has a special property called the Markov property. The Markov property means that the future state depends only on the current state, not on the sequence of events that preceded it.

Example: In simple words, "the future is independent of the past, given the present."

3. KL divergence：

KL divergence measures how different two probability distributions are. It is not symmetric, meaning KL(p||q) is not the same as KL(q||p).

Example: If you believe a coin is fair (50%-50%), but in reality it favors heads more, KL divergence measures how wrong your assumption is.

4. Carlo sampling method

Monte Carlo methods are techniques that use random sampling to estimate results, often when exact calculation is hard.

Example:

Throw a die many times to estimate the probability of rolling a 6, instead of calculating it exactly.

5. diffusion kernel：

A diffusion kernel is a way to measure similarity between points by simulating how heat would spread across them over time.

Example:

In a social network, a diffusion kernel can measure how "close" two users are through friends of friends.

6. Jensen’s inequality：

Jensen’s inequality says that for a convex function, the function of the average is less than or equal to the average of the function values.

Example:

If you have two numbers 2 and 8, the average of their squares is greater than the square of their average.

7. overfitting：

Overfitting happens when a model learns the training data too well, including its noise, and performs badly on new data.

Example:

You memorize the answers to math problems without understanding the methods, so you can't solve new problems.

8. posterior distribution：

The posterior distribution is the updated probability of a model or parameter after seeing new data.

Example:

When guessing the color ratio of balls in a bag, you adjust your guess after drawing several balls.

9. Gaussian kernel：

A Gaussian kernel measures similarity between two points based on a bell-shaped (normal) curve — closer points are more similar.

Example:

In image processing, a Gaussian kernel can blur a photo by smoothing nearby pixel colors.

10. multiply in closed form：

Multiplying in closed form means you can combine two expressions and get a final, simple formula without needing approximation.

Example:

Multiplying two Gaussian distributions results in another Gaussian distribution, and you can directly write down the new mean and variance.

2 Introduction of Diffusion Model

When we talk about diffusion model, most of people may think of Denoising Diffusion Probabilistic Models(DDPM),which is frequently applied for (un)conditional image/audio/video generation.This algorithm is widely applied by famous AI enterprises,such as Google Brain,OpenAI DELL and so on.
But in fact,it was proposed by Jascha S.D and so on in 2015.
Deep Unsupervised Learning using Nonequilibrium Thermodynamics has described the initial logic for us.Therefore, the author will share the basic thought of diffusion model by analyzing this article.So,let's get started.

2.1 Interpretation of Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Because the author only have poor English proficiency and a little academic aptitude,our article will keep the same order with the paper.

Abstract

First of all,let's read the abstract. It provides us with the central problem in machine learning, which is to model with highly flexible families of probability distributions for complex data-sets. During the process, we need to acknowledge that learning, sampling, inference and evaluation are still analytically or computationally tractable. However,generating data is an undeniable difficulty.Thus, the paper develop an approach that simultaneously achieves both flexibility and tractability to generate.The essential idea is to systematically and slowly destroy structure in a data distribution,which was inspired by non-equilibrium statistical physics. Then they learn a reverse diffusion process that restores structure in data, yielding a generative model. By this way,we can rapidly learn, sample from, and evaluate probabilities in deep generative models with thousands of layers or time steps, as well as to compute conditional and posterior probabilities under the learned model.

1.Introduction

Then it is the introduction. It share the difference between the tractability model and flexibility. As for models that are tractable can be analytically evaluated and easily fit to data. However,these failed to aptly describe structure in rich datasets. On the other hand, the flexible models can be molded to fit structure in arbitrary data. For the example it gave, it shows us the easy way to yield the flexible distribution and the difficult way to compute the essential constant in it. And then,the paragraph lists the development in this field.

1.1 Diffusion probabilistic models

In this part, the writer give us a novel way to define probabilistic models. It lists four key properties.

1. extreme flexibility in model structure,
2. exact sampling,
3. easy multiplication with other distributions,
4. the model log likelihood, and the probability of individual states, to be cheaply evaluated.

We need to thinking about the reason why are them.

1. **extreme flexibility:** The method uses a Markov chain to gradually transform one distribution into another. This allows the model to capture data distributions of arbitrary forms without requiring a predefined, non-analytically-normalizable potential function. The flexibility comes from the ability to define the model as the endpoint of the Markov chain, with each step of the chain analytically evaluable.
2. **exact sampling:** By explicitly defining the probability models as the endpoint of the Markov chain, the models allows for exact sampling. That is because the confirmed endpoint means the confirmed

calculation process, which avoids the approximation and randomness. This is crucial for achieving high accuracy in generative models, especially when working with complex datasets.

3. **easy multiplication with other distributions:** One of the key benefits of this approach is that it allows for straightforward multiplication with other distributions. This is particularly useful when performing tasks like computing posterior distributions or inpainting images, as the model can easily combine with additional information, such as noisy data.
4. **efficient evaluation of log-likelihood and state probabilities:** This point is essentially a repetition of the above content. Since each step in the diffusion process is analytically tractable, evaluating the model log-likelihood or the probability of individual states is computationally efficient. This is essential for optimizing the model and performing tasks like model evaluation and posterior inference.

Therefore, we succeeded to analysis the key logic under these four definitions. Let's continue reading below.

The next position has talked about the method with more details. The writer shares the expecial way to build a Markov chain, which called it generative Markov chain. In this chain, a simple known distribution was converted into a target(data) distribution by diffusion process. Instead of approximately evaluating a model which has been otherwise defined, they defined a probabilistic emodel as the endpoint of Markov chain. Due to the analytically evaluable probability, the full chain can also be analytically evaluated.

It is worth noting that the learning in this framework involves estimating small perturbations to a diffusion process, which is more tractable than explicitly describing the full distribution with a single, non-analytically-normalizable, potential function. Furthermore, since a diffusion process exists for any smooth target distribution, this method can capture data distributions of arbitrary form.

And to demonstrate the utility of these diffusion probabilistic models, the team also train high log likelihood models for several classic datasets.

1.2 Relationship to other work

In this paragraph, the writer shares the history of related research and shows the differences and advantages relative to these techniques:

1. **The development of framework using ideas from physics, quasi-static process, and annealed importance sampling rather from variational Bayesian methods:** to understand this change's advantage, we need to know the difference between annealed importance sampling and variational Bayesian methods.

AIS is Carlo sampling method. The process starts with a simple distribution(often uniform or prior) and slowly transitions via a temperature parameter to the target distribution. At each step of annealing process, AIS uses importance sampling. The importance weights are updated as the distribution changes. During the quasi-static process which AIS mimics, a system is driven slowly enough that it stays in or near equilibrium at all times.This helps that the sampling is more likely to represent the target distributions as the temperature decreases.

As for Variational Bayesian Methods ,they are optimization-based approaches to appoximate posterior distributions. In variational inference, a parameterized family of distributions is closen, and the algorithm aims to find the parameters of this distribution that minimize the KL divergence between the true posterior and the variational distribution. During the process of finding the best approximation within the chosen family of distributions, variational methods prefer to approximate complex posteriors by simpler, more tractable distributions. Although variational methods don't have a direct connection to quasi-static process, the methods attempt to minimize the divergence between the true posterior and approximation, essentially finding the "minimum energy" configuration for the distribution.
Therefore, we can realize that AIS will obtain a more direct approach.

2. **We show how to easily multiply the learned distribution with another probability distribution (e.g with a conditional distribution in order to compute a posterior):**

Firstly, we know that most of the generative models(like GAM、VAE、GSN), it is so difficult that multiplied the learned distribution with another distribution.Just like the last formula:

$$\tilde{p}(x) \propto p(x)r(x)$$

For other models, *p(x)* are usually not parsing expressions, so that we need to very complicated ways to get the samples. At the same time, the overall distribution doesn't have the original structure which means it will fail to generate data. However, diffusion model will generate it easily. It is based on Markov chain, which means we can restore the data step by step. At the same time, we can adjust the sampling direction as well. Thus, we can achieve the correction of conditional distribution. In this process, the core way is to add the correction factor *r(x)*. After that, we will compete our target by guided diffusion. It just like provide a slight force to guide the result to walk to the way what we want. Regarding the specific steps, we will have a detailed discussion on it in 2.5.

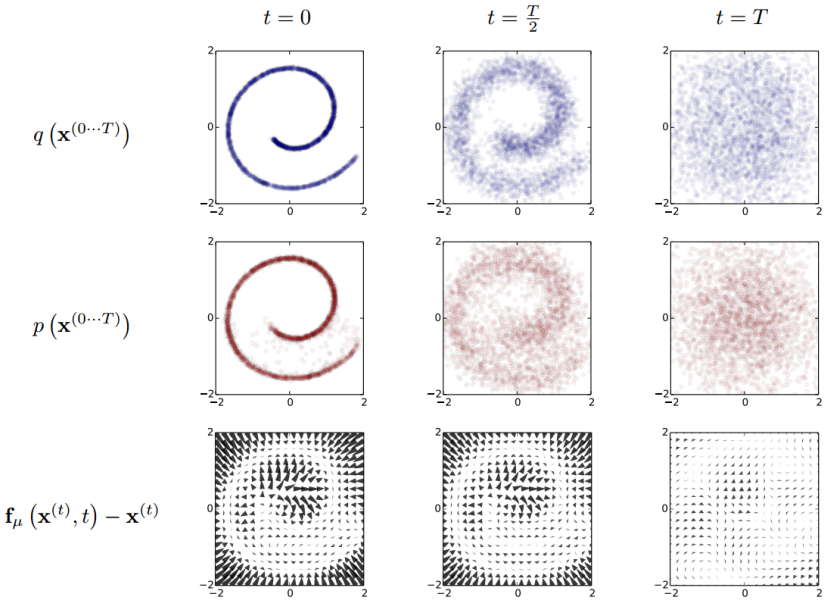
3. **We address the difficulty that training the inferencemodel can prove particularly challenging in varia-tional inference methods, due to the asymmetry in theobjective between the inference and generative models. We restrict the forward (inference) process to a simple functional form, in such a way that the re-verse (generative) process will have the same functional form:** this point is easier to understand. By a simple functional form, the reserse generative process will have the same sample function form. It is useful for calculations.
4. **We train models with thousands of layered, rather than only a handful of layers :** Due to the simple functional form, we'll enjoy a easy implementation of multiple count calculations.
5. **We provide upper and lower bounds on the entropy production in each layer(or time step):** By comparing the change of entropy, we can realize the increasing or decreasing uncertainty as the model transform data. Additionally, bounding the entropy, the model ensures that it doesn't produce unrealistic outputs.

Afterwards,the writer write a number of related technologies for training probabilistic models and explain the mathematical principles.

2.Algorithm

In this part, the writer show the detailed introduction of the various process of the algorithm.

During the process, we will define a forward(or inference) diffusion process which converts complex data distribution into a simple, tractable, distribution, and then learn a finite -time reversal of this diffusion process which defines our generative model distribution. The writer provides a figure to help understand.



In this picture, it briefly introduces the overall process. On 2-d swiss roll data, we trained this model. In the top row, they show the process of Gaussian diffusion. In the middle, it's the reverse trajectory of corresponding. By learned mean and covariance functions, we get the data distribution. In the last row, it shows the drift term,

$$f_\mu(x^{(t)}, t) - x^{(t)},$$

for the same reverse diffusion process.

Next, they show how the reverse, generative diffusion process can be trained and used to evaluate probabilities. In addition, they derive entropy bounds and show how the learned distributions can be multiplied by any second distribution.(The writer will prove a detailed explanation of the process in the corresponding section.)

2.1 Forward Trajectory

By repeating application of Markov diffusion kernel

$$T_\pi(y|y';\beta)$$

for

$$\pi(y)$$

where

$$\beta$$

is the diffusion rate, the data distribution is gradually converted into a well behaved distribution.

$$\pi(y) = \int dy' T_\pi(y|y';\beta)\pi(y') \qquad (1)$$
$$q(x^{(t)}|x^{(t-1)};\beta_t) = T_\pi(x^{(t)}|x^{(t-1)};\beta_t). \qquad (2)$$

Thus, the forward trajectory which is corresponded at the data distribution and performing T steps of diffusion is

$$q(x^{(0...T)}) = q(x^{(t)} \prod_{t=1}^{T} q(x^{(t)}|x^{(t-1)})$$

For the experiment shown below, it means either Gaussian diffusion into a Gaussian distribution with identity-covariance, or binomial diffusion into an independent binomial distribution. And Table 1 gives the diffusion kernels for both Gaussian and binomial distributions.

Dataset	K	$K - L_{null}$
Swiss Roll	2.35 bits	6.45 bits
Binary Heartbeat	-2.414 bits/seq.	12.024 bits/seq.
Bark	-0.55 bits/pixel	1.5 bits/pixel
Dead Leaves	1.489 bits/pixel	3.536 bits/pixel
CIFAR-10[3]	5.4 ± 0.2 bits/pixel	11.5 ± 0.2 bits/pixel
MNIST	See table 2	

Table 1. The lower bound K on the log likelihood, computed on a holdout set, for each of the trained models. See Equation 12. The right column is the improvement relative to an isotropic Gaussian or independent binomial distribution. L_{null} is the log likelihood of $\pi\left(\mathbf{x}^{(0)}\right)$. All datasets except for Binary Heartbeat were scaled by a constant to give them variance 1 before computing log likelihood.

As for Figure 2 and Figure 3, they show the complete training processes.

2.2 Reverse Trajectory

The generative distribution will be trained to describe the same same trajectory in reverse. The formula is shown below :

$$p(x^{(T)}) = \pi(x^{(T)}) \qquad (4)$$
$$p(x^{(0...T)}) = p(x^{(T)} \prod_{t=1}^{T} p(x^{(t-1)}|x^{(t)})$$

According to previous research, it is known that the continuous diffusion (limit of small step size) of both Gaussian and binomial diffusion has the identical functional form as the forward process. Thus, the function (3) will be the Gaussian distribution, which means the longer the trajectory the smaller the diffusion rate beta can be made.

During learning,we only need to estimate the mean and covariance for a Gaussian diffusion kernel, or the bit flip probability for a binomial kernel,which is a easier way to get what we want, just like the Table App 1 shown.

2.3 Model Probability

We need to know that the probability the generative model assigns to the data is

$$p(x^{(0)}) = \int dx^{(1...T)} p(x^{(0...T)}). \qquad (6)$$

We need to know that this integral is intractable. However, we are reminded that instead evaluate the relative probability of the forward and reverse trajectories, averaged over the forward trajectories by annealed importance sampling and the Jarzynski equality. Immediately after, the writer shown the derivation process.

$$p\left(x^{(0)}\right)=\int dx^{(1\cdots T)}p\left(x^{(1\cdots T)}\middle|x^{(0)}\right)\frac{q\left(x^{(1\cdots T)}\middle|x^{(0)}\right)}{q\left(x^{(1\cdots T)}\middle|x^{(0)}\right)} \quad (7)$$

$$=\int dx^{(1\cdots T)}q\left(x^{(1\cdots T)}\middle|x^{(0)}\right)\frac{p\left(x^{(0\cdots T)}\right)}{q\left(x^{(1\cdots T)}\middle|x^{(0)}\right)} \quad (8)$$

$$=\int dx^{(1\cdots T)}q\left(x^{(1\cdots T)}\middle|x^{(0)}\right)\cdot p\left(x^{(T)}\right)\prod_{t=1}^{T}\frac{p\left(x^{(t-1)}\middle|x^{(t)}\right)}{q\left(x^{(t)}\middle|x^{(t-1)}\right)} \quad (9)$$

In this process, we have successfully convert this integral into a simple form. By this way, the forward trajectory can be evaluated rapidly. At the same time, due to the infinitesimal beta, the forward and the reverse distribution can be made identical, which helps that a exactly evaluation can be obtained with only a single sample, as can be seen by substitution.

2.4 Training

Just like the writer say, training amounts to maximizing the model log likelihood, which has a lower bound provided by Jensen's inequality.

$$=\int dx^{(1\cdots T)}q\left(x^{(1\cdots T)}\middle|x^{(0)}\right)\cdot p\left(x^{(T)}\right)\prod_{t=1}^{T}\frac{p\left(x^{(t-1)}\middle|x^{(t)}\right)}{q\left(x^{(t)}\middle|x^{(t-1)}\right)} \quad (9)$$

$$L=\int dx^{(0)}q\left(x^{(0)}\right)\log p\left(x^{(0)}\right) \quad (10)$$

$$=\int dx^{(0)}q\left(x^{(0)}\right)\cdot\log\left[\int dx^{(1\cdots T)}q\left(x^{(1\cdots T)}\middle|x^{(0)}\right)\cdot\frac{p\left(x^{(T)}\right)}{q\left(x^{(1\cdots T)}\middle|x^{(0)}\right)}\prod_{t=1}^{T}\frac{p\left(x^{(t-1)}\middle|x^{(t)}\right)}{q\left(x^{(t)}\middle|x^{(t-1)}\right)}\right] \quad (11)$$

$$L\geq\int dx^{(0\cdots T)}q\left(x^{(0\cdots T)}\right)\cdot\log\left[p\left(x^{(T)}\right)\prod_{t=1}^{T}\frac{p\left(x^{(t-1)}\middle|x^{(t)}\right)}{q\left(x^{(t)}\middle|x^{(t-1)}\right)}\right] \quad (12)$$

Next, the writer shows the reduce process. It is worth nothing that if the forward and reverse trajectories are identical , the inequality in Equation 12 becomes an equality for the qual-static prcess.

$$L\geq K$$

$$K=-\sum_{t=2}^{T}\int dx^{(0)}dx^{(t)}q\left(x^{(0)},x^{(t)}\right)\cdot \quad (13)$$

$$D_{KL}\left(q\left(x^{(t-1)}\middle|x^{(t)},x^{(0)}\right)\|p\left(x^{(t-1)}\middle|x^{(t)}\right)\right)$$

$$+H_q\left(X^{(T)}\middle|X^{(0)}\right)-H_q\left(X^{(1)}\middle|X^{(0)}\right)-H_p\left(X^{(T)}\right) \quad (14)$$

Also, in the training content, we need to find the reverse Markov transitions which maximize this lower bound on the likelihood. The specific targets of estimation for Gaussian and binomial diffusion are given in Table App 1.

Thus, the task of estimating a probability distribution has been reduced to the task of performing regression on the functions which set the mean and covariance of sequence of Gaussian. By this transforming complex probability estimation tasks into regression tasks(fitting tasks), training becomes more efficient. This simplification makes it more feasible to train diffusion models on high-dimensional data and deep networks, and enables them to run effectively on large-scale datasets.

2.4.1 Setting The Diffusion Rate

We need to know that the choice of beta in the forward trajectory is important for the performance of the trained model. Reasonable schedule will improve the accuracy of log partition function estimate. And it makes a difference to the lost energy of the schedule taken when moving between equilibrium distributions.
In addition, the writer shows their special way to processing data. He pointed that they learn the forward diffusion schedule by gradient ascent on K. At the same time, the beta one is fixed to a small constant to prevent overfitting and the 'frozen noise' which means setting it as a constant is treated as an additional auxiliary variable. Besides, due to its discrete state space noise will inevitably change, we choose to erase a constant fraction 1/T of original signal per diffusion step so that we can yield the diffusion rate of beta t.

2.5 Multiplying Distributions, and Computing Posteriors

In this section, the writer shows the method of multiplying distributions using diffusion models. This also demonstrates a huge advantages of diffusion model.

As is known to all, there are a lot of tasks like computing a posterior in order to do signal denoising or inference of missing values need to multiply model distribution with a second distribution, or bounded positive function, producing a new distribution. However, it is so costly and difficult for many techniques, including variational autoencoders, GSNs, NADEs, and most graphical models. But for diffusion model, it is straightforward, since the second distribution can be treated ethier as a small perturbation to each step in the diffusion process, or often exactly multiplied into each diffusion step. Figure 3 and Figure 5 show this process.

In this figure, diffusion model is used to generated samples. Although the effect at this time is average, it still shares the feasibility of diffusion model.

In this figure, we can feel the powerful ability of diffusion model. After using a diffusion probability model trained on images of bark, we can get the realistic result.

However, how can we get the sample? The following content provides us with a answer.Original text:

2.5.1 Modified Marginal Distributions

In the section, we show how to modify the marginal distribution. To compute modified trajectory function, we multiply each of the intermediate distributions by a corresponding function r(x^(0)), which is its bounded positive function and denote this conversion process with a tilde above. It starts at the distribution below and proceeds through the sequence of intermediate distributions. the related function is that:

$$\tilde{q} = \frac{1}{Z_t} q(x^{(t)}) r(x^{(t)}), \qquad (16)$$

where Z is the normalizing constant for the t th intermediate distribution.

2.5.2 Modified Diffusion steps

In order to obtain powerful diffusion steps, we change the equilibrium condition. First, we know that the Markov kernel q(x^t | x^{(t + 1)}) for the reverse diffusion process obeys the equilibrium condition:

$$q(x^{(t+1)}|x^{(t)})q(x^{(t)}) = q(x^{(t)}|x^{(t+1)})q(x^{(t+1)}). \qquad (17)$$

The new chain must instead satisfy:

$$\tilde{q}(x^{(t+1)}|x^{(t)})\tilde{q}(x^{(t)}) = \tilde{q}(x^{(t)}|x^{(t+1)})\tilde{q}(x^{(t+1)}). \qquad (18)$$

To satisfy equation 18, we set that:

$$\tilde{q}(x^{(t+1)}|x^{(t)}) \propto q(x^{(t+1)}|x^{(t)})r(x^{(t+1)}), \qquad (19)$$

$$\tilde{q}(x^{(t)}|x^{(t+1)}) \propto q(x^{(t)}|x^{(t+1)})r(x^{(t)}), \qquad (20)$$

So that p^{tilde{}}(x^{(t)}|x^{(t + 1)}) is modified in the corresponding fashion at the same time.

$$\tilde{p}(x^{(t)}|x^{(t+1)}) \propto p(x^{(t)}|x^{(t+1)})r(x^{(t)}), \qquad (21)$$

2.5.3 Applying r(x^{(t)})

We think that if r(x^{(t)}) is sufficiently smooth, the it can be treated as a small perturbation to the reverse diffusion kernel p(x^{(t)}|x^{(t + 1)}). In this case, the modified conditional distributions will have the same function with the original.

Besides, if r(x^{(t)}) can be multiplied with a Gaussian(or binomial) distribution in closed form, then it can be directly multiplied with the reverse diffusion kernel in close form ,and need not to be treated as a perturbation. This applies in the case where r(x^{(t)}) consists of a delta func-tion for some subset of coordinates.

2.5.4 Choosing r(x^{(t)})

Typically, r(x^{(t)}) should be chosen to change slowly over the course of the trajectory. For this paper, we chose it to be constant,

$$r(x^{(t)}) = r(x^{(0)}) \qquad (22)$$

Another choice is

$$r(x^{(t)}) = r(x^{(0)})^{\frac{T-t}{T}}$$

In this case, it make no difference to the starting distribution to the starting distribution for the reverse trajectory. The guarantees that drawing the sample from p^{tilde{}}(x^{(T)}) for the reverse trajectory remains straightforward.

2.6 Entropy of Reverse Process

To maintain the stability of the results, we place the upper and lower bounds on the entropy of each step in the reverse trajectory by the known forward process.

$$H_q(X^{(t)}|X^{(t-1)}) + H_q(X^{(t-1)}|X^{(0)}) - H_q(X^{(t)}|X^{(0)}) \leq H_q(X^{(t-1)}|X^{(t)}) \leq H_q(X^{(t)}|X^{(t-1)}) \qquad (23)$$

where both the upper and lower depend only on the conditional forward trajectory and can be analytically computed.

3.Experiments

In this section, we trained the model and prove the effectiveness.

By a variety of continuous datasets, and a binary dataset, we trained the model. In all cases, the objective function and gradient were computed using Theano and model training was with SFO.

3.1 Toy Problems

3.1.1 Swiss Roll

By radial basis function network, we succeed in building a two dimensional swiss roll distribution.

3.1.2 Binary Heartbeta Distribution

By training on simple binary sequences of length 20, where 1 occurs every 5th time bin, and the remainder of bins are 0, using a multilayer perceptron to create the Bernoulli rates. The learning was nearly perfect.

3.2 Images

In this section, we show several datasets which we trained on. By them, the ability of diffusion model can be proved.

3.2.1 Datasets

MNIST: we trained on MNIST digits and have a direct comparison against previous work. Besides, we estimate MNIST log likelihood using the Parzen-window code.

Model	*Log Likelihood*
Dead Leaves	
MCGSM	1.244 bits/pixel
Diffusion	**1.489 bits/pixel**
MNIST	
Stacked CAE	121 ± 1.6 bits
DBN	138 ± 2 bits
Deep GSN	214 ± 1.1 bits
Diffusion	$\mathbf{220 \pm 1.9}$ **bits**
Adversarial net	225 ± 2 bits

Table 2. Log likelihood comparisons to other algorithms. Dead leaves images were evaluated using identical training and test data as in (Theis et al., 2012). MNIST log likelihoods were estimated using the Parzen-window code from (Goodfellow et al., 2014), and show that our performance is comparable to other recent techniques.

CIFAR-10: A probabilistic model was fit to the training images for the CIFAR-10 challenge dataset. Samples from the trained model are provided in Figure 3.

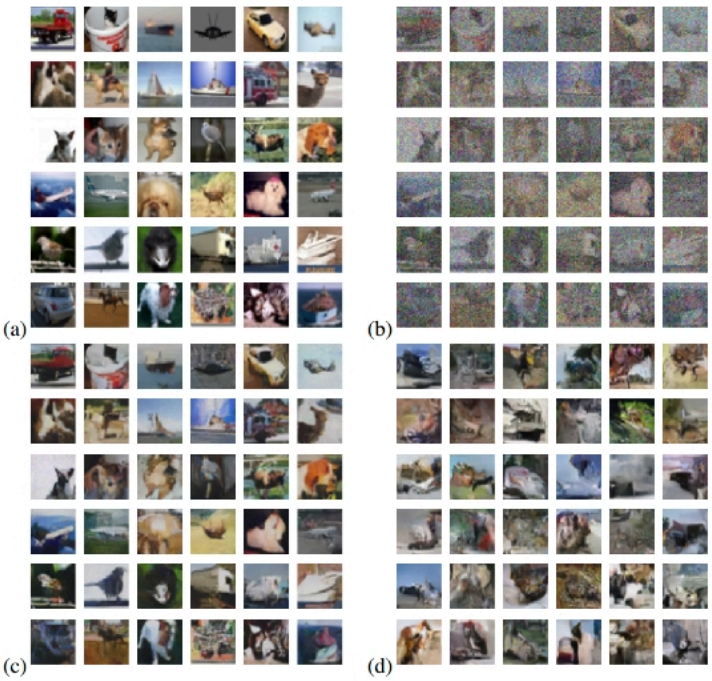


Figure 3. The proposed framework trained on the CIFAR-10 (Krizhevsky & Hinton, 2009) dataset. *(a)* Example holdout data (similar to training data). *(b)* Holdout data corrupted with Gaussian noise of variance 1 (SNR = 1). *(c)* Denoised images, generated by sampling from the posterior distribution over denoised images conditioned on the images in (b). *(d)* Samples generated by the diffusion model.

Dead Leaf Images: Its analytically tractable structure and many of the statistical complexities of natural images provide a compelling test case for nature images model. Figure 2 and 4 show the state of our model.

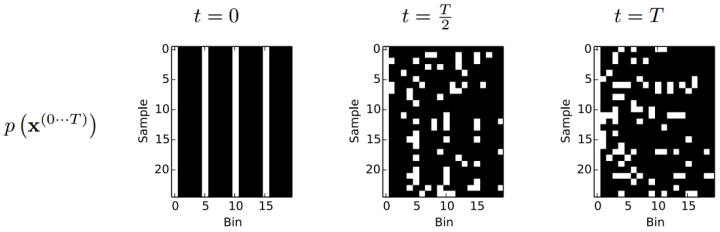


Figure 2. Binary sequence learning via binomial diffusion. A binomial diffusion model was trained on binary 'heartbeat' data, where a pulse occurs every 5th bin. Generated samples (left) are identical to the training data. The sampling procedure consists of initialization at independent binomial noise (right), which is then transformed into the data distribution by a binomial diffusion process, with trained bit flip probabilities. Each row contains an independent sample. For ease of visualization, all samples have been shifted so that a pulse occurs in the first column. In the raw sequence data, the first pulse is uniformly distributed over the first five bins.

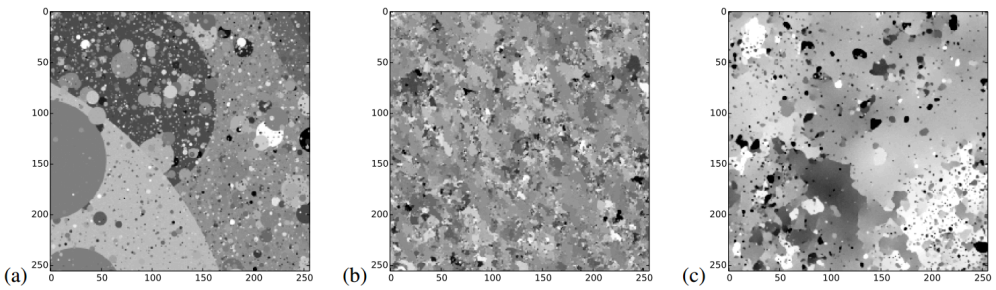


Figure 4. The proposed framework trained on dead leaf images (Jeulin, 1997; Lee et al., 2001). *(a)* Example training image. *(b)* A sample from the previous state of the art natural image model (Theis et al., 2012) trained on identical data, reproduced here with permission. *(c)* A sample generated by the diffusion model. Note that it demonstrates fairly consistent occlusion relationships, displays a multiscale distribution over object sizes, and produces circle-like objects, especially at smaller scales. As shown in Table 2, the diffusion model has the highest log likelihood on the test set.

Bark Texture Images: For this dataset we generate the missing data.

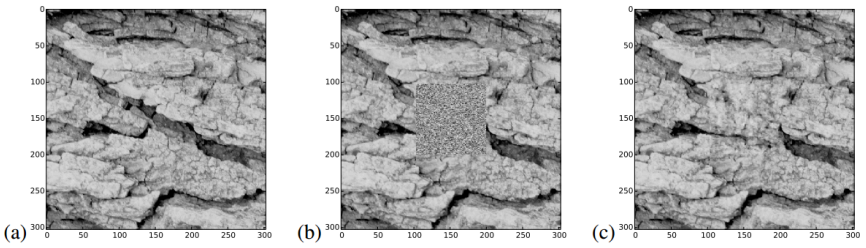


Figure 5. Inpainting. *(a)* A bark image from (Lazebnik et al., 2005). *(b)* The same image with the central 100×100 pixel region replaced with isotropic Gaussian noise. This is the initialization $\tilde{p}\left(\mathbf{x}^{(T)}\right)$ for the reverse trajectory. *(c)* The central 100×100 region has been inpainted using a diffusion probabilistic model trained on images of bark, by sampling from the posterior distribution over the missing region conditioned on the rest of the image. Note the long-range spatial structure, for instance in the crack entering on the left side of the inpainted region. The sample from the posterior was generated as described in Section 2.5, where $r\left(\mathbf{x}^{(0)}\right)$ was set to a delta function for known data, and a constant for missing data.

4.Conclusion

Overview of the entire text, we propose a novel algorithm for modeling probability distributions, which can exact simple and evaluate. By a lot of datasets, we use a similar basic algorithm to prove our model's effectiveness. It is worth nothing that most existing density estimation techniques must sacrifice modeling power to keep tractable and efficient with a extremely expensive sampling or evaluation.

The core of our algorithm is Markov chain. Each diffusion step becomes simple and easy to estimate by mapping data to a noise distribution while the number of steps growing.

Thus, the result is a algorithm that can be suitable for any data distribution but which remains tractable to train, exactly sample form, and evaluate, and directly manipulate of conditions and posterior distribution.

Original text:

非平衡热力学进行深度无监督学习.pdf

Original text: