

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE CIENCIAS FÍSICAS

DEPARTAMENTO DE FÍSICA TEÓRICA



TRABAJO DE FIN DE GRADO

Código de TFG: FT33

Aprendizaje Automático Fuera del Equilibrio

Out-of-Equilibrium Machine Learning

Supervisor/es: Aurélien Decelle

Luis Bustinduy de la Guerra

Grado en Física

Curso académico 2023-24

Convocatoria Ordinaria de Junio

Resumen:

Los modelos de difusión han vivido en los últimos años una serie de transformaciones y desarrollos que los han convertido en herramientas de gran utilidad y muy competitivas en el ámbito de la computación generativa. Por lo tanto a través de este trabajo trataremos de estudiar los últimos desarrollos bajo un mismo marco y buscaremos aplicar de forma experimental las conclusiones teóricas alcanzadas, así como introducir una posible conexión con el estudio de las transiciones de fase.

El estudio se iniciará desde las bases de la termodinámica del no equilibrio, donde se entiende la difusión como el mecanismo de conducción consecuencia del cambio o gradiente en alguna de las propiedades de un sistema frente a una fuerza termodinámica. Al igual que hablamos de difusión de partículas o difusión del calor podemos hablar de difusión de la información en los sistemas. En este sentido buscaremos entonces estudiar cómo evoluciona la probabilidad frente a un proceso difusivo. Para lograrlo se tomará una cadena de difusión de Markov que transformará una muestra original de datos ordenada en $t = 0$ hacia una distribución gaussiana termalizada en $t = T$. De esta forma, estudiando la probabilidad del proceso buscaremos revertir su funcionamiento a través de un segundo proceso de difusión en este caso inverso.

Dicho proceso inverso es el que modelizaremos empleando una arquitectura lo más sencilla posible, que permita mantener una trazabilidad del aprendizaje pero lo suficientemente práctica como para obtener buenos resultados. Se emplearán autoencoders de una sola capa oculta.

Se comprobará entonces cómo el proceso inverso podrá llevar distribuciones gaussianas a la distribución original de nuestros datos y podremos generar así muestras en ese espacio totalmente nuevas.

Abstract:

In recent years, Diffusion Models have undergone a series of transformations and developments that have turned them into very useful and competitive tools in the field of generative computing. Consequently, this work studies the last developments of diffusion models under an unified frame, and constructs a practical application from the conclusions drawn. Additionally, a link to phase transitions is introduced.

The study will begin from the basis of non-equilibrium thermodynamics, where diffusion is understood as the conduction mechanism due to a change or gradient in a systems property under a thermodynamical force. As well as particle or heat diffusion, we can study diffusion occurring in the information of a system. Therefore, it would be convenient to study the evolution of probability through a diffusive process. To achieve this goal, a Markov Diffusion Chain will be used to transform an original data distribution $t = 0$ into a thermalized Gaussian distribution $t = T$. Hence, from this probability, a second and reversed diffusion process can be constructed.

This second reverse diffusion process will be the goal to modelize through the simplest yet practical machine learning architecture, single hidden layer autoencoders. That would allow a traceable study of the model together with good results.

From this study it will be proved how the reverse process can transform samples from Gaussian distributions back into new samples living in our original data distribution.

Índice

1. Introducción:	1
2. Difusión en física	2
2.1. Trayectoria forward y reverse:	3
3. Modelos de difusión y aprendizaje automático	4
3.1. Optimización	4
3.2. Proceso directo y truco de reparametrización:	5
3.3. Desarrollo de la función de pérdida	6
4. Algoritmo de entrenamiento	10
5. Implementación	11
5.1. Autoencoder	11
5.2. Resultados:	11
6. Transiciones de fase a través de modelos de difusión	15
7. Conclusión:	18
8. Anexo:	19
8.1. Anexo I: Paralelismo con el desarrollo de otros modelos de inferencia variacional	19
8.2. Anexo II: Perceptron y MNIST	20

1. Introducción:

Los modelos generativos son aquellas construcciones computacionales que habiendo sido entrenadas con una serie de datos pertenecientes a una distribución (como pueden ser imágenes de una cierta categoría, audio, datos de estructuras celulares, redes de spines...) buscan reconstruir la función de probabilidad original para poder tomar muestras y así crear ejemplos totalmente nuevos que vivan en la misma distribución estadística que nuestros datos originales.

Existen una gran variedad de modelos generativos probabilísticos pero los modelos de difusión desde su primera implementación en 2015 [14], han sido capaces de dar una respuesta de gran calidad al problema de modelar datos de alta dimensionalidad manteniendo un formulación probabilística abierta y trazable.

La trazabilidad es la capacidad de mantener un razonamiento analítico del entrenamiento y evaluación de un modelo. A día de hoy la complejidad en las arquitecturas de los modelos ha aumentado exponencialmente perdiendo en la mayoría de casos la posibilidad de un desarrollo analítico, pero escudándose en los excelentes resultados numéricos, es lo que se conoce como fenómeno de caja negra.

Esta pérdida de trazabilidad también ocurre en las versiones más avanzadas de modelos de difusión, dado que de forma comercial se introducen arquitecturas como UNet, ControlNet... en el entrenamiento de cara a la obtención de buenos resultados en espacios de alta dimensionalidad.

A través de este trabajo se desarrollarán los modelos de difusión de forma completa y se empleará una arquitectura trazable junto con las técnicas actuales de difusión con el objetivo de obtener resultados correctos en espacios de alta dimensión (MNIST), teniendo en cuenta los márgenes y limitaciones de emplear un arquitectura menos compleja.

Para ello en primer lugar será importante plantear las bases físicas sobre las cuales se fundamenta el desarrollo probabilístico que se construirá más adelante.

2. Difusión en física

La difusión desde el estudio de la termodinámica del no equilibrio consiste en la evolución de un sistema conforme a la ecuación:

$$\frac{\partial \mathbf{K}}{\partial t} = D \Delta \mathbf{K} \quad (1)$$

Conocida como ecuación de difusión donde Δ es el laplaciano y D la constante de difusión. En el contexto de difusión térmica esta expresión surge de la derivación de la ecuación de balance de la energía interna en combinación con la ley de Fourier que establece la relación entre el flujo de calor y el gradiente de la temperatura [6]. Pero existen otros procesos que también adoptan esta forma. Los random walks son uno de los ejemplos más representativos de procesos estocásticos que tienen como solución la ecuación de difusión. En el caso de los modelos de difusión lo que se busca es transformar una distribución inicial en otra a través de una evolución de no equilibrio. Para ello necesitamos un proceso de difusión que tenga una solución estacionaria y dado que (1) no nos ofrece este tipo de comportamiento se necesitará una modificación. Se empleará entonces un proceso de Ornstein-Uhlenbeck, el cual tiene asociado un potencial que es el responsable de la aparición de la solución estacionaria. Este proceso tiene una representación de densidad de probabilidad (en el espacio de Fokker-Planck) que toma la forma [12](5.23):

$$\frac{\partial \mathbf{K}(t, \mathbf{y}, \mathbf{y}')}{\partial t} = \theta \frac{\partial}{\partial \mathbf{y}'} (\mathbf{y} \mathbf{K}(t, \mathbf{y}, \mathbf{y}')) + D \frac{\partial^2}{\partial \mathbf{y}'^2} \mathbf{K}(t, \mathbf{y}, \mathbf{y}') \quad (2)$$

Donde $\mathbf{K}(t, \mathbf{x}, \mathbf{y})$ es el kernel de transición o kernel de calor que cumple [12](5.21):

$$\pi(\mathbf{y}) = \int d\mathbf{y}' \mathbf{K}(\mathbf{y}, \mathbf{y}', t) \pi(\mathbf{y}') \quad (3)$$

Pudiéndose interpretar la ecuación (1) como un caso libre de (2).

En nuestro caso, emplearemos un kernel de difusión de Markov $\mathbf{K}(\mathbf{y}, \mathbf{y}', t) = T_\pi(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}; \beta)$ que es una particularización de un proceso de Ornstein-Uhlenbeck visto a tiempos discretos [13] caracterizados mediante $t_T = -\frac{1}{2} \sum_{i=1}^T \log(\beta_t - 1)$ junto con $\theta = 1$ y $D = 1$.

Al ser un proceso de Markov buscamos que cada paso temporal dependa exclusivamente del anterior por lo que podemos definir $\mathbf{y} \rightarrow \mathbf{x}^{(t)}$ y $\mathbf{y}' \rightarrow \mathbf{x}^{(t-1)}$ y la dependencia temporal será caracterizada por β_t (Diffusion Rate). Se forma así el denominado paso forward [14]:

$$q\left(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}\right) = T_\pi\left(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}; \beta_t\right) \quad (4)$$

Que cumplirá entonces la propiedad de Markov:

$$q\left(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)} \dots \mathbf{x}^{(0)}\right) = q\left(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}\right) \quad (5)$$

El proceso de difusión de Markov que elegiremos para el paso forward con β_t como parámetro es el siguiente:

$$q\left(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}\right) = \mathcal{N}\left(\mathbf{x}^{(t)}; \mathbf{x}^{(t-1)} \sqrt{1 - \beta_t}, \mathbf{I} \beta_t\right) \quad (6)$$

Se trata de un proceso que añade una cierta cantidad de ruido en función del diffusion rate β_t a cada paso temporal. La aplicación reiterada de este paso llevará una distribución original $\mathbf{x}^{(0)}$ a un estado final difuso $\mathbf{x}^{(T)}$, formando una trayectoria forward.

En la sección final, correspondiente a la relación de las transiciones de fase con los modelos de difusión, se lleva a cabo una unión de esta dinámica con procesos de Langevin a través de funciones score y se estudian los potenciales correspondientes. Estas funciones no serán empleadas en el desarrollo principal que se expone a continuación (aunque surgen de forma natural en nuestro objetivo final).

2.1. Trayectoria forward y reverse:

El paso forward, aplicado reiteradamente, lleva nuestra distribución original a estados cada vez más difusos hasta llegar a un equilibrio a través de una trayectoria forward que tiene entonces la forma:

$$q(\mathbf{x}^{(0 \dots T)}) = q(\mathbf{x}^{(0)}) \prod_{t=1}^T q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) \quad (7)$$

La cual no es más que la composición de las distribuciones obtenidas tras los distintos pasos forward a cada tiempo desde $t = 0 \rightarrow t = T$. Completando una trayectoria forward habremos conseguido una distribución de datos que es puramente ruido, diremos que se llega a una termalización. Las características de la distribución final $\mathbf{x}^{(T)}$ comúnmente se eligen para llegar en el equilibrio a una distribución normal estandar $\mathcal{N}(\mathbf{0}, \mathbf{I})$, pero se pueden tomar y estudiar otras variedades como veremos en la sección de resultados.

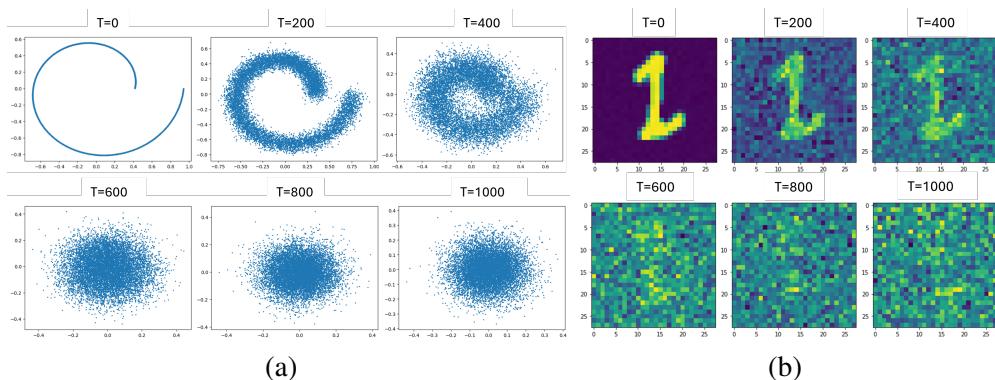


Figura 1: Ejemplos de procesos de difusión en los cuales se ha evaluado la trayectoria forward sobre dos datasets. En el estudio se emplearán datos de espirales (Swiss Roll) y de MNIST aquí presentados. El caso (a) presenta difusión a una distribución $\mathcal{N}(\mathbf{0}, (\mathbf{0}, \mathbf{1}))$ frente al caso (b) a $\mathcal{N}(\mathbf{0}, \mathbf{I})$

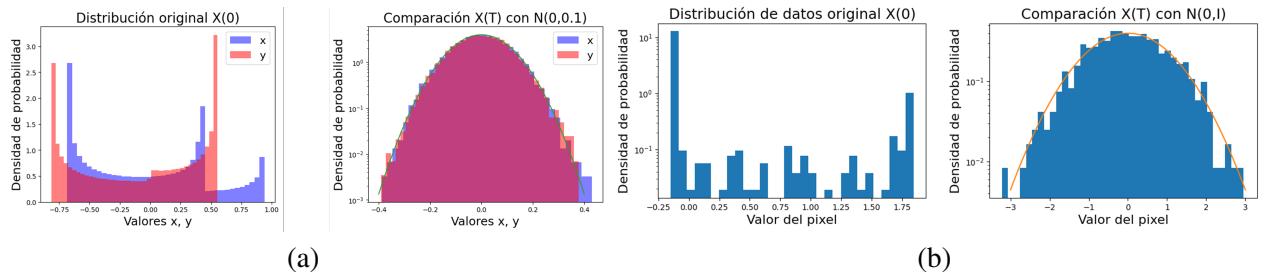


Figura 2: De los ejemplos anteriores se muestran los histogramas de los datos previos y posteriores al proceso de difusión para los datasets de espirales (a) (en todo el dataset) y MNIST (b) (en una sola muestra)

Se puede definir una trayectoria análoga pero inversa en el tiempo que también se tratará de un proceso de diffusión [1]. Se tratará del proceso reverse, también definido a través de una cadena de Markov, el cual parte de datos difusos $\mathbf{x}^{(T)}$ y los lleva a $\mathbf{x}^{(0)}$ siguiendo:

$$p(\mathbf{x}^{(0 \dots T)}) = p(\mathbf{x}^{(T)}) \prod_{t=1}^T p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) \quad (8)$$

Esta trayectoria es desconocida y sus parámetros característicos en cada paso serán la media y la varianza que denominaremos:

$$p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) = \mathcal{N}(\mathbf{x}^{(t-1)}; \mu(\mathbf{x}^{(t)}, t), \Sigma(\mathbf{x}^{(t)}, t)) \quad (9)$$

$$\mu(\mathbf{x}^{(t)}, t), \Sigma(\mathbf{x}^{(t)}, t)$$

3. Modelos de difusión y aprendizaje automático

Hasta este momento tenemos como resultado dos trayectorias. Por un lado la trayectoria forward, está totalmente definida ya que se trata del proceso difusivo de nuestra distribución original, y por otro lado, la trayectoria reverse, que se encuentra abierta a ser parametrizada y es la que busca revertir el proceso de difusión. La idea es conseguir un modelo que pueda elegir de forma idónea los parámetros de cada paso reverse para que tras llevar a cabo la trayectoria completa podamos obtener un resultado en $p(\mathbf{x}^{(0)})$ lo más cercano a $q(\mathbf{x}^{(0)})$, es decir un resultado que viva en la distribución de nuestros datos originales [1].

3.1. Optimización

Para conseguir el objetivo introducido necesitamos un parámetro que resuma el comportamiento del modelo y que podamos optimizar a través del entrenamiento. En los últimos años se ha llevado a cabo este estudio de ajuste de parámetros desde varios enfoques, en un primer momento se propuso optimizar el log likelihood:

$$l = \int d\mathbf{x}^{(0)} q(\mathbf{x}^{(0)}) \log(p(\mathbf{x}^{(0)})) \quad (10)$$

La cual veremos que se trata de una optimización computacionalmente exigente que llevaba a resultados competitivos en marcos específicos pero peor capacidad de ser generalizados.

Evaluar $p(\mathbf{x}^{(0)})$ directamente en $\mathbf{x}^{(0)}$ se trata de una tarea difícil por lo que conviene desarrollarla a lo largo del resto de la trayectoria reverse:

$$p(\mathbf{x}^{(0)}) = \int d\mathbf{x}^{(1\cdots T)} p(\mathbf{x}^{(0\cdots T)}) \quad (11)$$

Ahora añadimos el cociente de la trayectoria forward desde el paso $t = 1$ condicionada a la distribución inicial para llevar a cabo Importance Sampling [10], es decir, buscaremos evaluar $p(\mathbf{x}^{(0)})$ desde la más sencilla trayectoria $q(\mathbf{x}^{(1\cdots T)})$ que no conocemos pero podemos generar fácilmente a través del kernel de difusión de Markov. Éste término nos permitirá más adelante obtener de vuelta la trayectoria forward completa al evaluar el log likelihood y nos permitirá cambiar el enfoque del paso concreto en $q(\mathbf{x}^{(0)})$ a la trayectoria completa $q(\mathbf{x}^{(0\cdots T)})$.

$$p(\mathbf{x}^{(0)}) = \int d\mathbf{x}^{(1\cdots T)} p(\mathbf{x}^{(0\cdots T)}) \frac{q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)})}{q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)})}$$

$$= \int d\mathbf{x}^{(1\cdots T)} q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)}) \frac{p(\mathbf{x}^{(0\cdots T)})}{q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)})}$$

Dado que se ha impuesto que cada paso de la trayectoria depende únicamente del anterior se puede argumentar que:

$$q(\mathbf{x}^{(t=3)}, \mathbf{x}^{(t=2)}, \mathbf{x}^{(t=1)}) = q(\mathbf{x}^{(t=3)} | \mathbf{x}^{(t=2)}) q(\mathbf{x}^{(t=2)} | \mathbf{x}^{(t=1)}) q(\mathbf{x}^{(t=1)})$$

Por lo que la trayectoria que se inicia en $T = 1$ ($q(\mathbf{x}^{(1\cdots T)})$) en vez de en $T = 0$ ($q(\mathbf{x}^{(0\cdots T)})$) seguirá la forma:

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

Expandiendo también la trayectoria reverse y sacando el término T del productorio llegamos a la probabilidad de generar nuestra distribución original de datos de la forma:

$$p(\mathbf{x}^{(0)}) = \int d\mathbf{x}^{(1\cdots T)} q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)}) \cdot p(\mathbf{x}^{(T)}) \prod_{t=1}^T \frac{p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})}{q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)})} \quad (12)$$

y por lo tanto introduciéndolo de vuelta en la ecuación (10):

$$l = \int d\mathbf{x}^{(0)} q(\mathbf{x}^{(0)}) \log \left(q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)}) \cdot p(\mathbf{x}^{(T)}) \prod_{t=1}^T \frac{p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})}{q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)})} \right) \quad (13)$$

Una reducción resulta de evaluar un límite superior empleando la desigualdad de Jensen. Se trata de un teorema que demuestra: $\mathbf{E}[f(X)] \geq f(\mathbf{E}[X])$ y por lo tanto:

$$l \geq \int d\mathbf{x}^{(0)} q(\mathbf{x}^{(0\cdots T)}) \log \left(p(\mathbf{x}^{(T)}) \prod_{t=1}^T \frac{p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})}{q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)})} \right)$$

Podemos simplificar la notación si completamos la trayectoria reverse del productorio y volvemos a enmascarar la integral de la esperanza bajo el símbolo \mathbf{E} esta vez respecto a la trayectoria forward completa:

$$l \geq \mathbf{E}_{fw} \left[\log \left(\frac{p(\mathbf{x}^{(0\cdots T)})}{q(\mathbf{x}^{(1\cdots T)})} \right) \right] \quad (14)$$

Un modelo que consiga entrenar los parámetros de $p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})$ tendría que evaluar las dos trayectorias completas de cada muestra del dataset para poder minimizar la ecuación (14) y llevar a cabo una actualización de los pesos si se emplea una técnica de optimización como ascenso al gradiente.

Como habíamos anticipado se trata de un resultado con una gran carga computacional, existen formas de evaluarlo más rápidamente, pero la que obtuvo grandes resultados fue la propuesta en 2020 por Ho et al. [8] que replanteó el log likelihood y llevó a cabo este estudio desde las herramientas de la Evidence Lower Bound (ELBO).

Antes de entrar en contenido de este método vamos a definir el objetivo. Trataremos de obtener un parámetro optimizable, es decir, un objetivo que sea lo suficientemente resumido que permita una buena flexibilidad computacional y dependencias exclusivamente en parámetros relevantes de la difusión pero que garantice un buen funcionamiento.

Avanzaremos que se necesitará un proceso forward que esté a la altura de estas necesidades y que nos permita conseguir una difusión a un tiempo específico sin tener que recurrir a evaluar toda la secuencia temporal, necesitaremos un proceso forward directo.

3.2. Proceso directo y truco de reparametrización:

Lidiando con distribuciones de probabilidad la forma más efectiva de evaluar nuestro modelo de aprendizaje automático es hacerlo sacando la componente aleatoria a través de una variable ξ fuera del modelo. Esto se consigue reparametrizando la distribución. Para una variable que siga una distribución normal de la forma

$x \sim \mathcal{N}(\mu, \sigma^2)$ podemos expresarlo de la forma $x = \mu + \xi \cdot \sigma$ con $\xi \sim \mathcal{N}(0, 1)$ lo que se conoce como truco de reparametrización. Por lo que podemos escribir la distribución del paso reverse $q(x^{(t)} | x^{(t-1)})$ como:

$$x^{(t)} = \sqrt{1 - \beta^{(t)}} x^{(t-1)} + \sqrt{\beta^{(t)}} \xi \quad (15)$$

Esta sería la difusión que tendríamos que aplicar paso a paso generando variables cada vez más difusas iterando T veces a lo largo de la cadena de Markov. Pero si definimos medidas acumulativas de los parámetros siguiendo [8] podremos definir una reparametrización directa de la trayectoria completa:

$$\alpha^{(t)} = 1 - \beta^{(t)} \quad \bar{\alpha}^{(t)} = \prod_{s=1}^t \alpha_s$$

Si sustituimos las α en nuestra expresión obtenemos:

$$q(x^{(t)} | x^{(t-1)}) = \sqrt{\alpha^{(t)}} x^{(t-1)} + \sqrt{1 - \alpha^{(t)}} \xi$$

Si repetimos la reparametrización desde un paso de tiempo anterior $q(x^{(t)} | x^{(t-2)})$ obtendríamos:

$$x^{(t)} = \sqrt{\alpha^{(t)} \alpha^{(t-1)}} x^{(t-2)} + \sqrt{1 - \alpha^{(t)} \alpha^{(t-1)}} \xi$$

Y repetimos el proceso t veces $q(x^{(t)} | x^{(0)})$ llegaremos a:

$$x^{(t)} = \sqrt{\alpha^{(t)} \alpha^{(t-1)} \dots \alpha^{(1)} \alpha^{(0)}} x^{(0)} + \sqrt{1 - \alpha^{(t)} \alpha^{(t-1)} \dots \alpha^{(1)} \alpha^{(0)}} \xi$$

O lo que es equivalente:

$$x^{(t)} = \sqrt{\bar{\alpha}^{(t)}} x^{(0)} + \sqrt{1 - \bar{\alpha}^{(t)}} \xi \quad (16)$$

Por lo que ahora teniendo la $\bar{\alpha}^{(t)}$ de nuestro paso t deseado, en una sola iteración de nuestros datos podemos llegar al estado difuso en el paso t.

3.3. Desarrollo de la función de pérdida

La función de Evidence Lower Bound (ELBO) se construye como el límite inferior a la función de pérdida por lo tanto si cambiamos el signo y sentido de nuestro resultado anterior (14):

$$l \leq -\mathbf{E}_{fw} \left[\log \left(\frac{p(x^{(0 \dots T)})}{q(x^{(1 \dots T)})} \right) \right] = \mathbf{E}_{fw} \left[\log \left(\frac{q(x^{(1 \dots T)})}{p(x^{(0 \dots T)})} \right) \right] = ELBO \quad (17)$$

Las técnicas de inferencia variacional que se emplean a modelos de difusión no son específicas, sino que las comparten una familia de modelos de variables latentes y pueden cubrirse desde un formalismo más general que se introduce en el Anexo I.

Siguiendo con nuestros resultados paralelamente a lo desarrollado en [16], lo que nos queda es por un lado la trayectoria forward partiendo de un conjunto inicial de datos $q(x^{(1 \dots T)} | x^{(0)})$ y por otro lado la trayectoria reverse en el denominador $p(x^{(0 \dots T)})$

Si desarrollamos las trayectorias con las fórmulas (7) y (8) y sacamos el término $p(x^{(t)})$ del logaritmo llegamos a:

$$\mathbf{E}_{fw} \left[-\log(p(x^{(T)})) + \log \left(\frac{\prod_{t=1}^T q(x^{(t)} | x^{(t-1)})}{\prod_{t=1}^T p(x^{(t-1)} | x^{(t)})} \right) \right]$$

Los productorios los podemos sacar del logaritmo como sumatorios:

$$\mathbf{E}_{fw} \left[-\log(p(\mathbf{x}^{(T)})) + \sum_{t=1}^T \log \left(\frac{q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)})}{p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})} \right) \right]$$

Ahora de cara a un posterior desarrollo se saca un término de la suma:

$$\mathbf{E}_{fw} \left[-\log(p(\mathbf{x}^{(T)})) + \sum_{t=2}^T \log \left(\frac{q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)})}{p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})} \right) + \log \left(\frac{q(\mathbf{x}^{(1)} | \mathbf{x}^{(0)})}{p(\mathbf{x}^{(0)} | \mathbf{x}^{(1)})} \right) \right]$$

El paso forward usando bayes nos queda:

$$q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) = \frac{q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) \cdot q(\mathbf{x}^{(t)})}{q(\mathbf{x}^{(t-1)})}$$

Será conveniente condicionar la expresión extendida del forward step con nuestra primera distribución de datos (la conocida) esto es una herramienta para dejar los datos difusos que generamos todavía asociados a nuestra primera distribución:

$$\frac{q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)}) \cdot q(\mathbf{x}^{(t)} | \mathbf{x}^{(0)})}{q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(0)})}$$

Lo llevamos a nuestra fórmula:

$$\mathbf{E}_{fw} \left[-\log(p(\mathbf{x}^{(T)})) + \sum_{t=2}^T \log \left(\frac{q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)}) \cdot q(\mathbf{x}^{(t)} | \mathbf{x}^{(0)})}{p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) \cdot q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(0)})} \right) + \log \left(\frac{q(\mathbf{x}^{(1)} | \mathbf{x}^{(0)})}{p(\mathbf{x}^{(0)} | \mathbf{x}^{(1)})} \right) \right]$$

Podemos separar términos del logaritmo:

$$\mathbf{E}_{fw} \left[-\log(p(\mathbf{x}^{(T)})) + \sum_{t=2}^T \log \left(\frac{q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)})}{p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})} \right) + \sum_{t=2}^T \log \left(\frac{q(\mathbf{x}^{(t)} | \mathbf{x}^{(0)})}{q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(0)})} \right) + \log \left(\frac{q(\mathbf{x}^{(1)} | \mathbf{x}^{(0)})}{p(\mathbf{x}^{(0)} | \mathbf{x}^{(1)})} \right) \right]$$

El tercer término desarrollado va a ir cancelando a lo largo de la suma todos los componentes menos el último en el numerador y el primero en el denominador quedando:

$$\mathbf{E}_{fw} \left[-\log(p(\mathbf{x}^{(T)})) + \sum_{t=2}^T \log \left(\frac{q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)})}{p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})} \right) + \log \left(\frac{q(\mathbf{x}^{(T)} | \mathbf{x}^{(0)})}{q(\mathbf{x}^{(1)} | \mathbf{x}^{(0)})} \right) + \log \left(\frac{q(\mathbf{x}^{(1)} | \mathbf{x}^{(0)})}{p(\mathbf{x}^{(0)} | \mathbf{x}^{(1)})} \right) \right]$$

Al expandir los logaritmos de los dos últimos términos se nos cancelará el denominador del penúltimo con el numerador del último quedando:

$$\mathbf{E}_{fw} \left[-\log(p(\mathbf{x}^{(T)})) + \sum_{t=2}^T \log \left(\frac{q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)})}{p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})} \right) + \log \left(q(\mathbf{x}^{(T)} | \mathbf{x}^{(0)}) \right) + \log \left(p(\mathbf{x}^{(0)} | \mathbf{x}^{(1)}) \right) \right]$$

Juntando primer y tercer término:

$$\mathbf{E}_{fw} \left[-\log \left(\frac{q(\mathbf{x}^{(T)} | \mathbf{x}^{(0)})}{p(\mathbf{x}^{(T)})} \right) + \sum_{t=2}^T \log \left(\frac{q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)})}{p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})} \right) + \log(p(\mathbf{x}^{(0)} | \mathbf{x}^{(1)})) \right]$$

Finalmente podemos poner nuestro resultado como divergencias KL $D_{KL}(p || q) = \int_x p(x) \log \frac{p(x)}{q(x)} dx$:

$$\boxed{l \leq \mathbf{E}_{fw} \left[D_{KL} \left(q(\mathbf{x}^{(T)} | \mathbf{x}^{(0)}) || p(\mathbf{x}^{(T)}) \right) + \sum_{t=2}^T D_{KL} \left(q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)}) || p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) \right) - \log(p(\mathbf{x}^{(0)} | \mathbf{x}^{(1)})) \right]} \quad (18)$$

El siguiente paso será llevar a cabo una serie de reducciones y aproximaciones empezando por despreciar:

- La primera divergencia KL no va a ser tenida en cuenta, el motivo es que estamos comparando la trayectoria forward que lo único que hace es añadir ruido y no contiene parámetros de nuestro ajuste; y por otro lado la distribución ya difusa que es ruido homogéneo.
- El tercer término toma valores grandes si en el salto de tiempo de 1 a 0 en la trayectoria reverse los puntos próximos son bien aproximados por el modelo, y toma valores bajos si en ese salto de tiempo no se acerca a la distribución original. La forma de no tenerlo en cuenta es si en la trayectoria reverse no añadimos ruido en el último paso y de esa forma tendremos un término estable en ese paso de tiempo que podemos despreciar de la loss function.

Continuaremos desarrollando entonces el término intermedio el cual es una suma de divergencias KL donde aparecen dos distribuciones que trataremos de reducir.

La segunda distribución es la trayectoria reverse la cual contiene los parámetros que queremos entrenar $p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) = \mathcal{N}(\mathbf{x}^{(t-1)}; \mathbf{f}_\mu(\mathbf{x}^{(t)}, t), \mathbf{f}_\Sigma(\mathbf{x}^{(t)}, t))$. Y la primera una nueva distribución condicionada también por $\mathbf{x}^{(0)}$ a la que le asignamos la forma, $q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)}) = \mathcal{N}(\mathbf{x}^{(t-1)}; \tilde{\mathbf{f}}_\mu(\mathbf{x}^{(t)}, t), \tilde{\mathbf{f}}_\Sigma(\mathbf{x}^{(t)}, t))$

Si usamos Bayes podemos expresar esta distribución como:

$$q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}, \mathbf{x}^{(0)}) = \frac{q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}, \mathbf{x}^{(0)}) \cdot q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(0)})}{q(\mathbf{x}^{(t)} | \mathbf{x}^{(0)})}$$

Que puede ser expandido en la forma gaussiana de las distribuciones de la forma:

$$\exp \left(-\frac{1}{2} \left(\frac{(\mathbf{x}^{(t)} - \sqrt{\alpha^{(t)}} \mathbf{x}^{(t-1)})^2}{\beta^{(t)}} + \frac{(\mathbf{x}^{(t-1)} - \sqrt{\bar{\alpha}^{(t-1)}} \mathbf{x}^{(0)})^2}{1 - \bar{\alpha}^{(t-1)}} - \frac{(\mathbf{x}^{(t)} - \sqrt{\bar{\alpha}^{(t)}} \mathbf{x}^{(0)})^2}{1 - \bar{\alpha}^{(t)}} \right) \right)$$

Si expandimos los cuadrados y combinamos los términos $\mathbf{x}^{(t-1)}$ que será la variable que queremos para nuestra distribución encontramos que los índices del coeficiente $(\mathbf{x}^{(t)})$ corresponderán con la varianza de la distribución buscada y los que acompañan al término $(\mathbf{x}^{(t)})^2$ se identificarán con su media obteniendo entonces:

$$\tilde{\mu}^{(t)}(\mathbf{x}^{(t)}, \mathbf{x}^{(0)}) = \frac{\sqrt{\bar{\alpha}^{(t-1)}} \beta^{(t)}}{1 - \bar{\alpha}^{(t)}} \mathbf{x}^{(0)} + \frac{\sqrt{\alpha^{(t)}} (1 - \bar{\alpha}^{(t-1)})}{1 - \bar{\alpha}^{(t)}} \mathbf{x}^{(t)} \quad \tilde{\beta}^{(t)} = \frac{1 - \bar{\alpha}^{(t-1)}}{1 - \bar{\alpha}^{(t)}} \beta^{(t)}$$

Si nos enfocamos en la media podemos cambiar $\mathbf{x}^{(t)}$ por la expresión reparametrizada que sacamos en la sección anterior (16) que despejando nos queda:

$$\mathbf{x}^{(0)} = \frac{1}{\sqrt{\bar{\alpha}^{(t)}}} \left(\mathbf{x}^{(t)} - \sqrt{1 - \bar{\alpha}^{(t)}} \xi \right)$$

Si lo ponemos en la expresión de nuestra media:

$$\tilde{\mu}^{(t)}(\mathbf{x}^{(t)}, \mathbf{x}^{(0)}) = \frac{\sqrt{\bar{\alpha}^{(t-1)}}\beta^{(t)}}{1-\bar{\alpha}^{(t)}} \frac{1}{\sqrt{\bar{\alpha}^{(t)}}} \left(\mathbf{x}^{(t)} - \sqrt{1-\bar{\alpha}^{(t)}}\xi \right) + \frac{\sqrt{\alpha^{(t)}}(1-\bar{\alpha}^{(t-1)})}{1-\bar{\alpha}^{(t)}} \mathbf{x}^{(t)}$$

Teniendo en cuenta las expresiones de beta y alpha acumulativa: $\alpha^{(t)} = 1 - \beta^{(t)}$ y $\bar{\alpha}^{(t)} = \prod_{s=1}^t \alpha_s$ Podemos ir reduciendo nuestra expresión de la forma:

$$\mathbf{x}^{(t)} \left(\frac{\sqrt{\alpha^{(t)}}(1-\bar{\alpha}^{(t-1)})}{1-\bar{\alpha}^{(t)}} + \frac{\sqrt{\bar{\alpha}^{(t-1)}}\beta^{(t)}}{1-\bar{\alpha}^{(t)}} \frac{1}{\sqrt{\bar{\alpha}^{(t)}}} \right) - \frac{\sqrt{\bar{\alpha}^{(t-1)}}\beta^{(t)}}{1-\bar{\alpha}^{(t)}} \frac{\sqrt{1-\bar{\alpha}^{(t)}}\xi}{\sqrt{\bar{\alpha}^{(t)}}}$$

Si ponemos todos los términos como $\alpha^{(t)} = \frac{\bar{\alpha}^{(t)}}{\bar{\alpha}^{(t-1)}}$ tendremos:

$$\mathbf{x}^{(t)} \left(\frac{\sqrt{\alpha^{(t)}} - \sqrt{\alpha^{(t)}}\bar{\alpha}^{(t)}\frac{1}{\bar{\alpha}^{(t)}}}{1-\bar{\alpha}^{(t)}} + \frac{\sqrt{\bar{\alpha}^{(t)}}(1-\alpha^{(t)})}{(1-\bar{\alpha}^{(t)})\sqrt{\bar{\alpha}^{(t)}}\alpha^{(t)}} \right)$$

Juntamos denominador y eliminamos términos para llegar a nuestra expresión final:

$$\mathbf{x}^{(t)} \left(\frac{(\sqrt{\alpha^{(t)}} - \bar{\alpha}^{(t)}\frac{1}{\sqrt{\alpha^{(t)}}})\sqrt{\alpha^{(t)}} + (1-\alpha^{(t)})}{(1-\bar{\alpha}^{(t)})\sqrt{\alpha^{(t)}}} \right) = \mathbf{x}^{(t)} \frac{1}{\sqrt{\alpha^{(t)}}}$$

Ahora el segundo sumando de forma similar lo reformulamos en función del paso t:

$$-\frac{\sqrt{\bar{\alpha}^{(t)}}\beta^{(t)}}{1-\bar{\alpha}^{(t)}} \frac{\sqrt{1-\bar{\alpha}^{(t)}}\xi}{\sqrt{\alpha^{(t)}}\sqrt{\bar{\alpha}^{(t)}}} = -\frac{1}{\sqrt{\alpha^{(t)}}} \frac{\beta^{(t)}}{\sqrt{1-\bar{\alpha}^{(t)}}}\xi$$

Y podemos juntar los términos reduciéndose entonces nuestra expresión para la media a:

$$\tilde{\mu}^{(t)}(\mathbf{x}^{(t)}, \mathbf{x}^{(0)}) = \frac{1}{\sqrt{\alpha^{(t)}}} \left(\mathbf{x}^{(t)} - \frac{\beta^{(t)}}{\sqrt{1-\bar{\alpha}^{(t)}}}\xi \right)$$

La divergencia D_{KL} entonces nos comparará ambas distribuciones. Al ser gaussianas la divergencia KL se corresponderá con el cálculo del error cuadrático medio de ambas distribuciones:

$$L_t = \mathbf{E}_{fw} \left[\frac{1}{2\sigma_t^2} \| \tilde{\mu}^{(t)}(\mathbf{x}^{(t)}, \mathbf{x}^{(0)}) - \mu^{(\theta)}(\mathbf{x}^{(t)}, t) \|^2 \right]$$

Sustituimos $\tilde{\mu}^{(t)}$:

$$L_t = \mathbf{E}_{fw} \left[\frac{1}{2\sigma_t^2} \| \frac{1}{\sqrt{\alpha^{(t)}}} \left(\mathbf{x}^{(t)} - \frac{\beta^{(t)}}{\sqrt{1-\bar{\alpha}^{(t)}}}\xi \right) - \mu^{(\theta)}(\mathbf{x}^{(t)}, t) \|^2 \right]$$

El término $\mu^{(\theta)}(\mathbf{x}^{(t)}, t)$ que corresponde con la media de la trayectoria reverse que estamos comparando se puede formular a través de el truco de reparametrización de forma que dependa de la variable aleatoria $\mathbf{x}^{(t)}$ y de un ruido $\xi^{(\theta)}(\mathbf{x}^{(t)}, t)$ que sea el que entrenemos a través de nuestra red neuronal. La forma que toma la media y que tendremos que sustituir en el error cuadrático medio es $\mu^{(\theta)}(\mathbf{x}^{(t)}, t) = \mu^{(t)} = \frac{1}{\sqrt{\alpha^{(t)}}} \left(\mathbf{x}^{(t)} - \frac{\beta^{(t)}}{\sqrt{1-\bar{\alpha}^{(t)}}}\xi^{(\theta)}(\mathbf{x}^{(t)}, t) \right)$. Y juntando todo el problema se reducirá a:

$$L_t = \mathbf{E}_{fw} \left[\frac{(\beta^{(t)})^2}{2\sigma_t^2 \alpha^{(t)} (1-\bar{\alpha}^{(t)})} \| \xi - \xi^{(\theta)}(\mathbf{x}^{(t)}, t) \|^2 \right] \quad (19)$$

Por lo que vemos la función de pérdida queda totalmente simplificada y nos queda una comparación entre el ruido aplicado y el ruido predicho por el modelo, por lo que el entrenamiento consistirá en minimizar esta cantidad. En ocasiones se discute incluso la posibilidad de no tener en cuenta los términos proporcionales:

$$L_t = \mathbf{E}_{fw} \left[\| \xi - \xi^{(\theta)}(\mathbf{x}^{(t)}, t) \|^2 \right] \quad (20)$$

Este fue un resultado experimental en [8], pero que sigue un razonamiento específico. Dicho término se simplifica ya que aporta mayor peso a los términos de más bajo t los cuales son menos difusos, por lo que ampliar los pesos y por lo tanto la carga en el modelo para tiempos de mayor t permite un mayor enfoque en tiempos más difusos y según los autores mejores resultados. Estudiaremos este enfoque con una comparación en nuestra arquitectura.

Entonces nuestra distribución para el modelo será:

$$\mathcal{N} \left(\mathbf{x}^{(t-1)}; \frac{1}{\sqrt{\alpha^{(t)}}} \left(\mathbf{x}^{(t)} - \frac{\beta^{(t)}}{\sqrt{1 - \bar{\alpha}^{(t)}}} \xi^{(\theta)}(\mathbf{x}^{(t)}, t) \right), \beta^{(t)} \right)$$

Por lo tanto reparametrizando el paso reverse:

$$\mathbf{x}^{(t-1)} = \frac{1}{\sqrt{\alpha^{(t)}}} \left(\mathbf{x}^{(t)} - \frac{\beta^{(t)}}{\sqrt{1 - \bar{\alpha}^{(t)}}} \xi^{(\theta)}(\mathbf{x}^{(t)}, t) \right) + \sqrt{\beta^{(t)}} \xi \quad (21)$$

Pero como vimos antes para poder despreciar el tercer término de la función de pérdida (18), necesitaremos tomar la forma sin ruido añadido a nuestra distribución para el paso $t=1$:

$$\mathbf{x}^{(0)} = \frac{1}{\sqrt{\alpha^{(1)}}} \left(\mathbf{x}^{(1)} - \frac{\beta^{(1)}}{\sqrt{1 - \bar{\alpha}^{(1)}}} \xi^{(\theta)}(\mathbf{x}^{(1)}, t) \right) \quad (22)$$

4. Algoritmo de entrenamiento

Vista la forma de obtener las variables a lo largo de la trayectoria reverse, podemos entonces definir un algoritmo de entrenamiento que nos permita aprender $\xi^{(\theta)}(\mathbf{x}^{(t)}, t)$ de nuestra muestra de datos. Así luego podremos aplicar la trayectoria reverse a samples de ruido nuevo no visto durante el entrenamiento para generar distribuciones a través de nuestro modelo. [8] propone:

Algorithm 1 Training

- 1: repeat
 - 2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
 - 3: $t \sim \text{Uniform}(\{1, \dots, T\})$
 - 4: $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
 - 5: Take mini-batch gradient descent step on

$$\nabla_{\theta} \left(\omega_t \|\xi - \epsilon_{\theta} (\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \xi, t)\|^2 \right)$$
 - 6: until converged
-

Algorithm 2 Sampling

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

Tabla 1: Algoritmo de entrenamiento y evaluación propuesto por Ho et al. 2020

Una representación esquemática del entrenamiento será útil para ilustrar el algoritmo. En la siguiente figura se muestra un paso de actualización de pesos del algoritmo para un solo punto y tiempo del dataset:

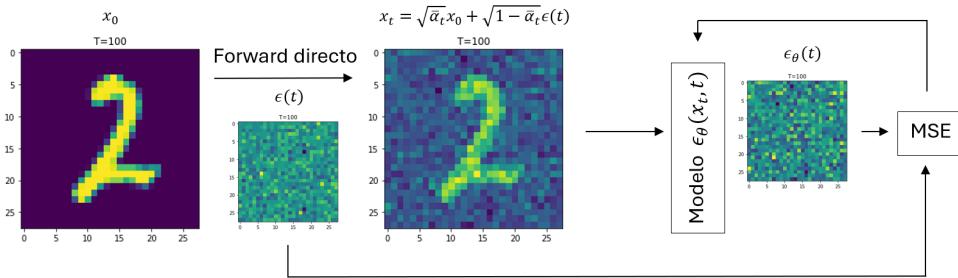


Figura 3: Esquema del algoritmo de entrenamiento para una iteración de actualización de pesos del modelo.

5. Implementación

A lo largo de la sección práctica se buscará implementar el modelo de difusión a través de una arquitectura lo más sencilla posible con el objetivo de mantener la trazabilidad del modelo, es decir, que exista la posibilidad de tener un desarrollo probabilístico de los pesos de nuestro sistema. Queremos que esto sea así dado que la mayor parte de implementaciones comerciales de difusión emplean grandes modelos (como UNet más comúnmente) conocidos como de caja negra, cuyo desarrollo probabilístico se pierde ante la complejidad de las capas y variables del modelo.

En el Anexo II se explica una prueba con modelo lineal y cómo no es posible aplicarla al caso de difusión dado que no es capaz de recuperar en el proceso reverse distribuciones no gaussianas.

La opción que supone un paso adelante pero trazable es el uso de single hidden layer autoencoders.

5.1. Autoencoder

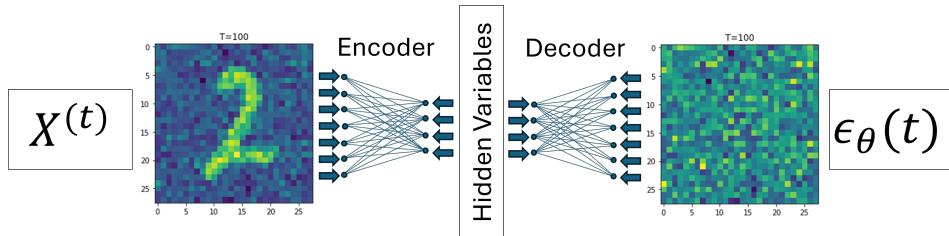


Figura 4: Esquema práctico del modelo de aprendizaje empleado. Se ilustra cómo los autoencoders transforman codifican y decodifican la dimensionalidad de la muestra.

Los autoencoders son arquitecturas que codifican la información de entrada a través de una capa de variables ocultas para luego decodificarla en el espacio de salida $\epsilon_\theta^{(t)}$. Tenemos entonces pesos que relacionan cada nodo del modelo el cual es capaz de entender estructuras no lineales.

La activación de los pesos, es decir, cómo se reduce del espacio funcional al espacio de fases del modelo, se lleva a cabo a través de la combinación de una función lineal de entrada, una ReLU de activación y una función lineal de salida.

Para parametrizar el tiempo dentro del modelo se puede introducir t como variable pero vamos a optar por definir un autoencoder para cada paso de tiempo que será entrenado por separado.

5.2. Resultados:

Durante la fase experimental se buscó entender y optimizar los parámetros que definen el modelo así como la arquitectura del autoencoder. Al tratarse de un modelo sencillo perdemos en adaptabilidad por lo que es

necesario un ajuste eficaz de parámetros como:

- Parámetros del entrenamiento:

- **Learning rate:** se empleó como optimizador Adam, el cual permite bastante flexibilidad al adaptarse durante el entrenamiento pero es importante escalar este parámetro de cara a una convergencia precisa pero no lenta.
- **Épocas de entrenamiento:** Se trata del número de veces que el modelo va a procesar la totalidad de los datos, es decir la cantidad de veces que se repetirá el algoritmo. Es crucial una buena elección para asegurar que el entrenamiento ha llegado a una convergencia.
- **Hidden variables:** correspondiente al tamaño del autoencoder, este parámetro tiene un gran papel en la capacidad del modelo de representar la realidad o sobreajustarse a los datos a los que ha sido expuesto.
- **Batch Size:** se empleó minibatch, esto significa que a la hora de actualizar los pesos se toma la media del gradiente de un conjunto de datos en vez de actualizar para cada punto del dataset, esto mejora la velocidad pero puede sacrificar la capacidad de generalización en caso de ser tomado muy alto.
- **Sample y test size:** tamaño de la muestra de datos empleada para el entrenamiento y tamaño de la muestra de test sobre la que se efectuarán pruebas para comprobar que no existe overfitting. En nuestro caso se emplearon las herramientas de compilación en CUDA de las librerías empleadas. Esto significa que los datos eran tomados por la tarjeta gráfica la cual operaba en paralelo con ellos, permitiendo entonces el uso fluido de grandes cantidades de datos para el entrenamiento.

- Parámetros de difusión:

- **Tiempo de difusión T:** el número de pasos en los que se realizará la difusión, Se necesita lo suficientemente alto para describir correctamente la difusión de forma discreta.
- **Varianza de la parametrización ξ :** Definirá la distribución a la que llegaremos tras la difusión.
- **Diffusion rate $\beta^{(t)}$:** Se trata de la sucesión de la varianza de nuestro proceso forward y será la que nos garantice una termalización de nuestros datos tras la difusión.

	Espiral 2π	Espiral 3π	MNIST
Difusión			
T	1000	1000	
ξ	0,1	1	1
$\beta^{(t)}$	$\beta^{(0)} = 1e - 4$ $\beta^{(T)} = 0,02$		$\beta^{(0)} = 1e - 4$ $\beta^{(T)} = 0,02$
Entrenamiento			
lr	0,01	0,005	$t = 0 \dots 4: 0,0001$ Resto: 0,0002
Épocas	20	30	300
Hidden Var	900	1000	1700
Batch	100	50	50
Sample y Test size	Sample: 10000	Sample: 20000	Sample 12500 Test: 2100

Tabla 2: Parámetros específicos del modelo para los datasets de espiral y MNIST a través de los cuales se han obtenido mejores resultados.

Se estudiarán dos datasets, espirales y MNIST [5]. Ambos representados junto con su proceso forward en las figuras (1b) (1a) pero en ellas se muestra una difusión de menor profundidad de la que emplearemos en el entrenamiento. En el caso de MNIST se entrenarán modelos con muestras de dos tipos de cifras diferentes cada vez, para así mejorar la velocidad de entrenamiento y mantener una exigencia más controlada del autoencoder. En la tabla (2) se resumen los parámetros empleados para el entrenamiento de la modalidad de resultados más satisfactoria.

Se han elegido estos datasets ya que suponen una muy buena forma de poner a prueba el modelo. Por un lado la modalidad de espiral/Swiss Roll se trata de datos bidimensionales muy específicos con los que comparar las características de aprendizaje de la forma de la distribución así como un proceso reverse con un objetivo muy estable. Por otra parte, MNIST supone una aplicación de alta dimensionalidad para el modelo (28x28 pixels).

Swiss Roll no permite ilustrar la fase generativa con tanto detalle ya que el espacio de nuestros datos (la espiral) está muy definido y la dimensionalidad es pequeña.

En la fase generativa de Swiss Roll se evalúan 1000 iteraciones de la trayectoria reverse bidimensional:

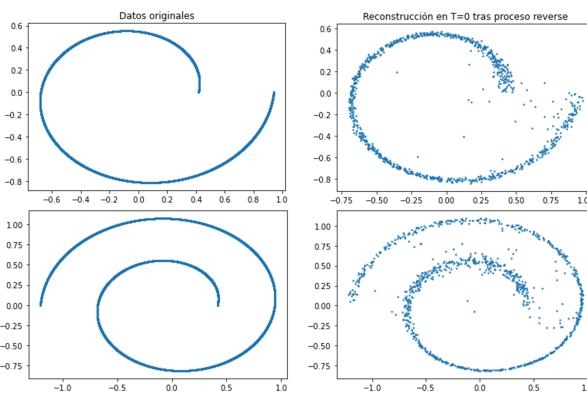


Figura 5: Datos experimentales producidos en la fase generativa del modelo de difusión entrenado con dataset de Swiss Roll para 2π arriba y 3π debajo, en el entrenamiento de 2π se empleó un batchsize de 100 y $\xi = 0,1$ frente al de 3π de 50 de batch y $\xi = 1$ para capturar en mayor detalle la zona interior.

A continuación, para **MNIST**, se ha empleado un mismo modelo entrenado en ejemplos de 0 y 1 en todo caso a excepción de (10c) y (6b) donde se han entrenado variaciones. La modalidad generativa con MNIST se puede aplicar evaluando un ruido gaussiano en la trayectoria reverse de dimensión correspondiente a los 784 píxeles de las imágenes:

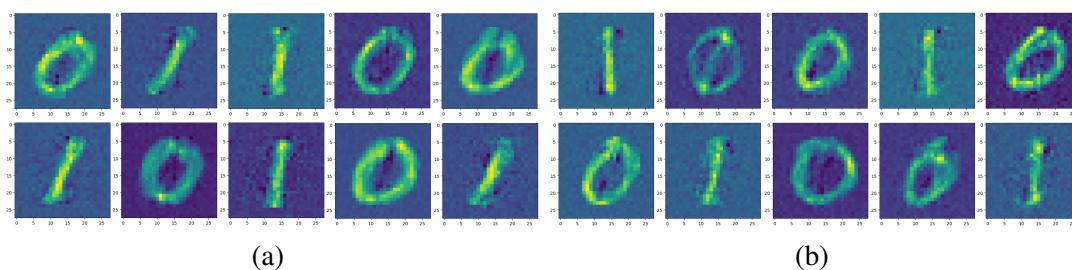


Figura 6: Serie de muestras generadas por el modelo a partir de un entrenamiento con cifras de 0 y 1 del dataset MNIST. Para (a) se ha empleado la esperanza completa teniendo en cuenta tanto los pesos como el MSE, y para (b) se ha empleado la versión reducida sin pesos.

Podemos entonces reconstruir muestras de la trayectoria reverse de MNIST para ilustrarla:

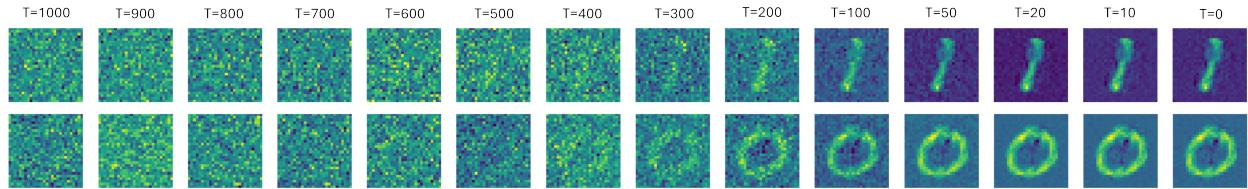


Figura 7: Trayectoria reverse de dos muestras de 0 y 1 de MNIST a distintos tiempos.

Se pueden estudiar otras aplicaciones de los modelos de difusión como es su uso en la reconstrucción de imágenes ruidosas. Para ello podemos recrear el proceso reverse desde un paso difuso de una muestra nunca antes vista por el modelo. Y se podrá estudiar hasta qué grado de difusión es posible una reconstrucción fiable en las circunstancias en las que se han entrenado nuestros modelos. Hay muchos más factores que pueden entrar en juego para mejorar este proceso de reconstrucción y dada la sencillez de nuestro modelo no se espera un resultado que lleve a estados cercanos a los datos originales pero sí que sea fiable, es decir nos devolverá un 0 o un 1 con las singularidades de nuestra muestra original junto con una considerable calidad.

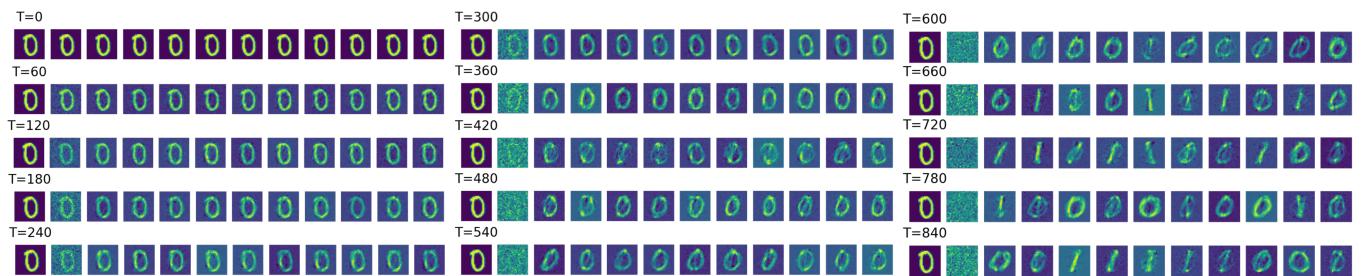


Figura 8: Sucesión de estados cada vez más difusos junto con 10 muestras de su reconstrucción, de izquierda a derecha la primera imagen es la original, la segunda el estado tras el proceso forward de difusión y las siguientes son las reconstrucciones.

Podemos ver que en torno a $T = 400$ conserva gran capacidad de reconstrucción pero empieza a cometer errores y a fallar en precisión. Y más adelante ya entramos en una pérdida de información que el modelo no es capaz de revertir. Otro aspecto es la mejora de calidad de imágenes donde el daño de los datos no es alto pero permite una mejora. Dos ejemplos son los siguientes:

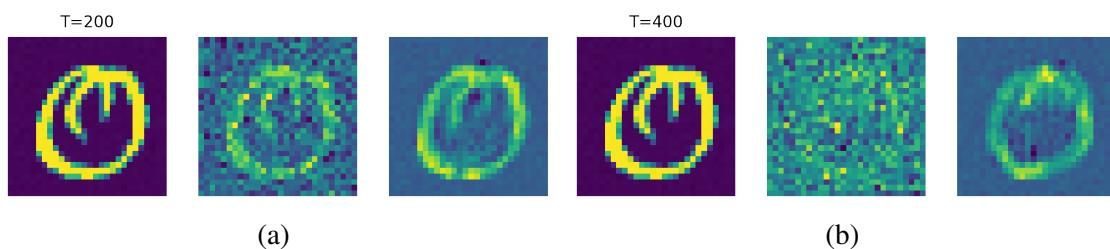


Figura 9: Ejemplos de mejora de calidad (a) y reconstrucción (b) empleando el proceso reverse sobre muestras de test difusas. En cada figura de izquierda a derecha se muestra primero la muestra de MNIST original, en segundo lugar el estado difuso obtenido tras un proceso reverse y que será del cual parta la reconstrucción, y en tercer lugar el resultado de la trayectoria reverse. Se puede observar la reconstrucción de detalles en etapas de alta difusión así como una buena capacidad de reducción de ruido en etapas de difusión baja.

Se estudió también la reconstrucción parcial de datos, es decir, se destruyen secciones de los datos originales y se evalúa la trayectoria reverse actualizando la zona dañada.

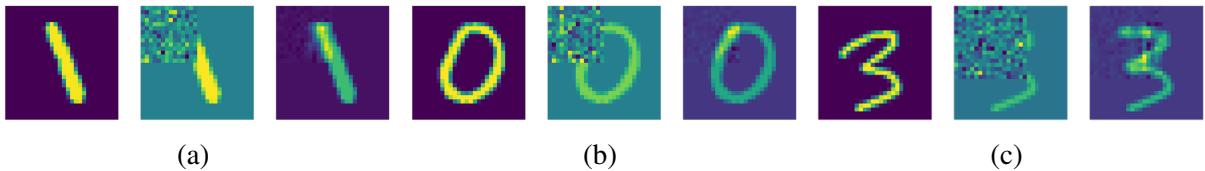


Figura 10: Reconstrucción de zonas parciales destruidas de los datos. De izquierda a derecha en cada muestra, la distribución original, el punto de partida dañado y el resultado en $T = 0$ de la trayectoria reverse. En (a) y (b) se emplea un modelo entrenado con cifras 0 y 1, y 14x14 píxeles destruidos. En (c) se empleó un modelo entrenado con cifras 1 y 3 y una destrucción de 18x18 más compleja.

Podemos ver resultados coherentes que son una muestra de la aplicabilidad de estas técnicas. En modelos comerciales estas capacidades se han desarrollado enormemente a través de enormes modelos capaces de mejorar la generación a etapas mucho más tempranas de difusión así como de estructuras más complejas y por lo tanto mejor definidas [15]. Pero los resultados presentados obtenidos mediante una arquitectura no compleja muestran la solidez estadística de la formulación de los modelos de difusión.

6. Transiciones de fase a través de modelos de difusión

Una pregunta que podemos hacernos de la capacidad generativa de los modelos de difusión es si en estos procesos puede haber una ruptura espontánea de simetría. Es decir, si los modelos pueden ser capaces de introducir dinámicas correspondientes a procesos como las transiciones de fase.

En el último año este tema ha estado a la orden del día y se han llevado a cabo las primeras relaciones teóricas entre modelos como Ising y la mecánica estadística de los modelos de difusión.

La forma de llevar a cabo estas relaciones ha sido a través de una generalización de la fase generativa de la difusión a grandes dimensiones.

Para mantener toda la generalidad del modelo vamos a tener que definir un formalismo de funciones score que englobe las probabilidades exactas de cada paso de la trayectoria forward con el objetivo de reconstruir dicha score en el paso reverse.

Entonces definimos primero las probabilidades exactas de los diferentes puntos de la trayectoria forward de la forma $q(\mathbf{x}^{(t)}, t) = q(\mathbf{x}^{(0)}) q(\mathbf{x}^{(t)} | \mathbf{x}^{(0)})$ por lo que desarrollando la distribución de forma gaussiana:

$$q(\mathbf{x}^{(t)}, t) = \int d\mathbf{x}^{(0)} q(\mathbf{x}^{(0)}) \frac{1}{\sqrt{2\pi(1-\bar{\alpha}^{(t)})^N}} \exp\left(-\frac{1}{2} \frac{\mathbf{x}^{(t)} - \mathbf{x}^{(0)} \sqrt{\bar{\alpha}^{(t)}}}{1-\bar{\alpha}^{(t)}}\right)$$

Se puede definir entonces una función score como [3]:

$$\mathbf{F}_i(\mathbf{x}^{(t)}, t) = \frac{\partial \log(q(\mathbf{x}^{(t)}, t))}{\partial \mathbf{x}_i^{(t)}} = -\frac{\mathbf{x}_i - \langle \mathbf{x}_i^{(0)} \rangle_{\mathbf{x}^{(t)}} \sqrt{\bar{\alpha}^{(t)}}}{1-\bar{\alpha}^{(t)}} = \xi \quad (23)$$

Donde $\langle \mathbf{x}_i^{(0)} \rangle_{\mathbf{x}^{(t)}}$ es el valor medio de $\mathbf{x}^{(0)}$ con respecto a la probabilidad condicionada $q(\mathbf{x}^{(0)} | \mathbf{x}^{(t)})$.

Con esta formulación de score el modelo de difusión se podría entrenar con una loss $l(\theta) = \mathbf{E} || \mathbf{S}(\mathbf{x}^{(t)}) - \mathbf{F}(\mathbf{x}^{(t)}, t) ||^2 + \mathbf{C}$. Donde \mathbf{S} es el modelo que a través de la esperanza va a tratar de aproximarse a \mathbf{F} , la función exacta de score. Vemos un gran paralelismo del resultado obtenido en (19) con la comparación de

funciones score y es que la resta entre un valor a cierta difusión y la media en $T = 0$ en (23) se puede entender como un ruido y que resultará exactamente en la forma (19).

Por otro lado, es importante remarcar que el proceso forward de difusión se puede entender como una evolución que sigue la ecuación de Langevin:

$$\frac{d\mathbf{x}^{(t)}}{dt} = -\mathbf{x}^{(t)} + \eta(t) \quad (24)$$

Si esto se cumple podemos decir que existe un proceso reverso de Langevin de la forma [1]:

$$-\frac{d\mathbf{y}_i}{dt} = +\mathbf{y}_i + 2\mathbf{F}_i(y, t) + \eta(t) \quad (25)$$

Que está mediado por una fuerza de sentido inverso a la de del proceso forward sumada a la función score [3]. Esto nos garantiza que siguiendo el proceso de reconstrucción se llegará a puntos $\mathbf{y}_i(t = 0)$ que estarán distribuidos de acuerdo a $q(t = 0)$.

La idea ahora es asociar exactamente la función score con el modelo de Ising para modelizar la evolución del parámetro de orden en campo medio a través de un proceso de difusión. Eso se conseguirá obteniendo la distribución $q(t = 0)$ del modelo de Ising y creando a distribución conjunta $q(\mathbf{x}^{(t)}, \mathbf{x}^{(0)})$. A través de ella podremos sacar un valor exacto de la función de score modelo S

Veremos cómo se puede aplicar al modelo de Curie-Weiss de interacción débil y alcance infinito de Ising construyendo muy resumidamente una función de probabilidad conjunta con difusión y centrándonos en el estudio del potencial. Curie-Weiss se define como:

$$q(\mathbf{x}^{(0)}) = \frac{1}{Z} \exp \left(\frac{\beta}{2N} \left(\sum_i \mathbf{x}_i^{(0)} \right)^2 + \frac{h}{N} \sum_i \mathbf{x}_i^{(0)} \right) \quad (26)$$

Donde $\mathbf{x}_i^{(0)}$ es un espín de la red que puede tomar los valores ± 1 .

Se buscará entonces una reparametrización del modelo en función de la magnetización siguiendo la transformación de Hubbard-Stratonovitch. Y empleando Steepest Descent en el límite de m continuo se conseguirá una distribución que en la fase ferromagnética ($m = \pm 1$ en equilibrio) presentará dos máximos, por lo que puede separarse la función de probabilidad de la forma:

$$q(m) = W_+ \delta(m - m^*) + W_- \delta(m + m^*) \quad m^* = \tanh(\beta m^*)$$

El siguiente objetivo es entonces obtener una distribución conjunta de $\mathbf{x}^{(t)}$ y $\mathbf{x}^{(0)}$ empleando los desarrollos anteriores y parametrizando debidamente en función de m.

$$q(\mathbf{x}^{(t)}, \mathbf{x}^{(0)}) = W_+ q^+(\mathbf{x}^{(t)}, \mathbf{x}^{(0)}) + W_- q^-(\mathbf{x}^{(t)}, \mathbf{x}^{(0)})$$

A través de esta probabilidad conjunta los autores aproximan la función $\langle \mathbf{x}_i^{(0)} \rangle_{\mathbf{x}^{(t)}}$ [3] que se trata de los valores medios de nuestros datos originales condicionados a los datos difusos que tengamos, obteniendo su desarrollo a través de toda la evolución temporal. Con ello se puede definir la función score:

$$\mathbf{S}(\mathbf{x}^{(t)})_i = -\frac{\mathbf{x}_i^{(t)}}{1 - \bar{\alpha}^{(t)}} + m^* \frac{\sqrt{\bar{\alpha}^{(t)}}}{1 - \bar{\alpha}^{(t)}} \tanh \left[m^* \left(fh + NM(\mathbf{x}^{(t)}) \frac{\sqrt{\bar{\alpha}^{(t)}}}{1 - \bar{\alpha}^{(t)}} \right) \right] \quad (27)$$

Siendo $\mathbf{M}(\mathbf{x}) = \frac{1}{N} \sum_i \mathbf{x}_i^{(t)}$ y expresando la variable $\mathbf{x}^{(t)}$ de forma que quede $\mu(t) = \left(\frac{1}{\sqrt{N}}\right) \sum_i \mathbf{x}_i^{(t)}$ podemos construir el potencial que cumplirá $-\frac{d\mu}{dt} = -\frac{dV}{d\mu} + \eta(t)$:

$$V = \frac{1}{2}\mu^2 - 2\log \left(\cosh \left[m^* \left(\mathbf{h} + \sqrt{N}\mu\sqrt{\bar{\alpha}^{(t)}} \right) \right] \right) \quad (28)$$

Del cual podemos extraer algunas conclusiones relativas al papel de la varianza.

- $\beta_t = 1 - \bar{\alpha}^{(t)}$ al ser la varianza de nuestro paso forward, es un parámetro que para tiempos cortos va a ser $\beta_{t \approx 1} \simeq 0$ lo que va a llevar a $\sqrt{\bar{\alpha}^{(t)}} \simeq 1$ y por lo tanto va a dominar el término \sqrt{N} . Pudiendo entonces aproximar el potencial a:

$$V_{\beta \approx 0}(\mu) = \frac{1}{2}\mu^2 - 2m^*\sqrt{N}\sqrt{\bar{\alpha}^{(t)}}|\mu|$$

Que se trata de un potencial con dos mínimos cuadráticos separados conforme al segundo término que es función del la varianza y por lo tanto del tiempo de difusión sobre el que nos encontramos.

- Mientras que para tiempos largos se espera que la varianza de la distribución forward sea $\beta_{t \gg 1} \simeq 1$ en el caso de que desemboquemos en una gaussiana estándar y por lo tanto $\sqrt{\bar{\alpha}^{(t)}} \simeq 0$ y se hará un término asintóticamente pequeño. Así pues se puede aproximar:

$$V_{\beta \approx 1}(\mu) = \frac{1}{2} \left[\mu - m^* \tanh(hm^*) \sqrt{N} \sqrt{\bar{\alpha}^{(t)}} \right]^2 + cte$$

Se trata de un potencial cuadrático desplazado. Y por lo tanto podemos comparar ambos potenciales:

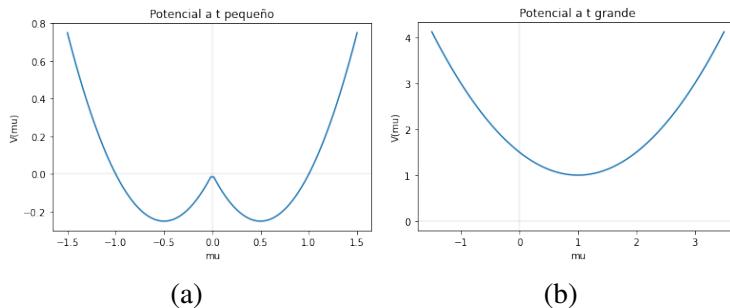


Figura 11: Potenciales para los casos extremos del término de la varianza

- Nos falta entonces otro régimen que es en el que el término $\sqrt{N}\sqrt{1 - \beta_t} \simeq 1$ lo cual ocurre para $\beta \simeq 1 - \frac{1}{N}$. Esta es la escala de tiempos (tendríamos que definir una función de evolución de β_t) en la cual se produce la transición entre los dos comportamientos expuestos del potencial para altos y bajos tiempos.

En esta escala de tiempos intermedia vemos que si volvemos a nuestra función score y sustituimos β tenemos un primer orden dominante en el término de la tanh proporcional a $\frac{1}{\sqrt{N}}$, por lo tanto esa debe ser nuestra escala de cálculo del score para poder distinguir ambas fases. Garantizado eso se demuestra entonces que nuestro modelo de difusión puede introducir la transición de fase dentro de su comportamiento generativo.

Vista esta posibilidad podemos considerar el desarrollo de la energía libre de una arquitectura como la que hemos empleado a lo largo del trabajo [9] de un autoencoder de una sola capa oculta con encoder ($H(W\mathbf{x} + b_h)$) y decoder $r(\mathbf{x}) = Ah(W\mathbf{x} + b_h) + b_r$ lineales habiendo nosotros elegido h como una

función ReLU introduciremos un cambio que es definirla como una función lineal de forma que se puede definir una energía potencial:

$$V_{linear}(\mathbf{x}) = \frac{1}{2} (\mathbf{W}\mathbf{x} + b_h)^T (\mathbf{W}\mathbf{x} + b_h) - \frac{1}{2} (\mathbf{x} + b_r)^2 \quad (29)$$

Que podemos asociar directamente al potencial de difusión obtenido ya que nuestro modelo emplea un autoencoder para cada tiempo de la cadena, abriendo las puertas a estudiar cómo evoluciona el aprendizaje de los pesos de la arquitectura en el potencial definido.

7. Conclusión:

Los modelos de difusión suponen una unión altamente práctica entre los fundamentos de la dinámica del no equilibrio y la inferencia variacional. El resultado de dicha unión hemos podido desarrollar a lo largo de este trabajo así como hemos podido aplicar de forma trazable. Los resultados obtenidos superan las expectativas teniendo en cuenta que se ha mantenido un uso exclusivo de autoencoders de una sola capa oculta, lo cual pone de manifiesto la solidez del desarrollo.

Quedan cuestiones abiertas en este estudio y que poco a poco están repercutiendo en nuestro día a día. La velocidad de los modelos se busca que sea cada vez mayor, la linea de investigación de la relación sencillez/calidad sigue abierta [11].

Se ha estudiado también a través de un ejemplo la forma en la que los modelos de difusión pueden llegar a describir rupturas de simetría motivándose así una futura línea de experimentación práctica.

Pero frente a todo ello, este trabajo explora las bases de la aplicabilidad de los modelos de difusión a autoencoders de una sola capa oculta, los cuales, son los únicos que pueden ser descritos analíticamente a través de su energía libre [9] a modo de función score. Es decir, el próximo paso a seguir es estudiar la evolución del aprendizaje a través de los propios pesos del modelo.

Referencias

- [1] Anderson, B.D. (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12, 313-326.
- [2] Andrew Ng and Tengyu Ma. June 11. (2023). CS229 Lecture Notes.
- [3] Biroli, Giulio & Mézard, Marc. (2023). Generative diffusion in very large dimensions.
- [4] Decelle, A. (2022). Fundamental problems in Statistical Physics XIV: Lecture on Machine Learning. *Physica A: Statistical Mechanics and its Applications*.
- [5] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6), 141–142.
- [6] García Villaluenga, Juan Pedro, y Armando Relaño Pérez. 2018. Termodinámica en sistemas fuera del equilibrio. Madrid: Ediciones Complutense.
- [7] Grigor, Alexander & Yan. (2005). Heat kernels on weighted manifolds and applications.
- [8] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. ArXiv, abs/2006.11239.
- [9] Kamyshanska, Hanna & Memisevic, Roland. (2013). On autoencoder scoring. 30th International Conference on Machine Learning, ICML 2013.
- [10] Neal, R.M. (2001). Annealed importance sampling. *Statistics and Computing* 11, 125–139.
- [11] Nichol, A., & Dhariwal, P. (2021). Improved Denoising Diffusion Probabilistic Models. ArXiv, abs/2102.09672.
- [12] Risken, H. (1989). The Fokker–Planck Equation: Methods of Solution and Applications.

- tions. New York: Springer-Verlag. ISBN 978-0387504988
- [13] Santos, J.E., & Lin, Y.T. (2023). Using Ornstein-Uhlenbeck Process to understand Denoising Diffusion Probabilistic Model and its Noise Schedules. ArXiv, abs/2311.17673.
- [14] Sohl-Dickstein, J.N., Weiss, E.A., Maheswaranathan, N., & Ganguli, S. (2015). Deep Unsupervised Learning using Nonequilibrium Thermodynamics. ArXiv, abs/1503.03585.
- [15] Song, Y., Dhariwal, P., Chen, M., & Sutskever, I. (2023). Consistency Models. International Conference on Machine Learning.
- [16] Weng, Lilian. (Jul 2021). What are diffusion models? Lil'Log.

8. Anexo:

8.1. Anexo I: Paralelismo con el desarrollo de otros modelos de inferencia variacional

En otros estudios de inferencia variacional de maximización de esperanza (como pueden ser los autoencoders variacionales o Mixture of Gaussians) se define la función ELBO como [2] $\text{ELBO}(x; Q, \theta) = \log(p(x)) - D_{KL}(Q(\mathbf{z}) \parallel p_{z|x})$. Esta ecuación recoge el límite inferior para un modelo cuyas variables visibles de salida son generadas por la distribución p que depende de unas variables ocultas del sistema \mathbf{z} , siendo Q una distribución de dichas variables ocultas. Es decir, existen una distribución de variables ocultas y no conocidas que van a definir la forma en la que se distribuyen nuestros datos visibles. Según el paralelismo de ambos desarrollos los modelos de difusión pueden ser interpretados como modelos de variable latente donde la distribución Q que va a generar nuestras latent variables será la función q forward por lo que las latent variables son $x^{(1\cdots T)}$, se elige condicionada a nuestros datos originales $x^{(0)}$. Es decir, elegimos como variables ocultas los puntos que viven entre nuestro estado difuso y nuestros datos originales, y será un caso más sencillo dado que la distribución Q la podemos elegir arbitrariamente independiente de los parámetros del modelo, mientras que en otros modelos de forma general esta es ajustada también dependiendo de ellos. Llevado a nuestro caso entonces:

$$\mathbf{E}_q \left(-\log(p(\mathbf{x}^{(0)})) \right) \leq \mathbf{E}_q(-\log(p(\mathbf{x}^{(0)})) + D_{KL}(q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)}) \parallel p(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)})))$$

Donde podemos desmontar las divergencias KL en su definición (primera igualdad) y sustituir el término inferior por Bayes (segunda igualdad):

$$\log \left(\frac{q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)})}{p(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)})} \right) = \log \left(\frac{q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)}) \cdot p(\mathbf{x}^{(0)})}{p(\mathbf{x}^{(0)} | \mathbf{x}^{(1\cdots T)}) \cdot p(\mathbf{x}^{(1\cdots T)})} \right)$$

El término inferior se trata de la probabilidad conjunta de los sucesos: $p(\mathbf{x}^{(0)}, \mathbf{x}^{(1\cdots T)})$ de esta forma vemos que tenemos de vuelta la trayectoria reverse, $p(\mathbf{x}^{(0\cdots T)})$ a su vez podemos separar términos:

$$\log \left(\frac{q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)}) \cdot p(\mathbf{x}^{(0)})}{p(\mathbf{x}^{(0\cdots T)})} \right) = \log \left(\frac{q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)})}{p(\mathbf{x}^{(0\cdots T)})} \right) + \log(p(\mathbf{x}^{(0)}))$$

Este término que nos sale va a ser el encargado de eliminar el de nuestra primera expresión ELBO para así recuperar nuestro límite inferior obtenido en el desarrollo principal a través de la desigualdad de Jensen:

$$\mathbf{E}_q \left(-\log(p(\mathbf{x}^{(0)})) \right) \leq \mathbf{E}_q \left(\log \left(\frac{q(\mathbf{x}^{(1\cdots T)} | \mathbf{x}^{(0)})}{p(\mathbf{x}^{(0\cdots T)})} \right) \right)$$

8.2. Anexo II: Perceptron y MNIST

Para entender el modelo utilizado en el proceso de difusión, vamos primero a mostrar algunas implementaciones útiles para entender el entrenamiento de una red neuronal [4].

Uno de los modelos históricos más importantes es el perceptrón, un algoritmo de clasificación. Su funcionamiento se basa en la activación de una serie de neuronas en función de una serie de pesos. Es decir, definiremos una serie de parámetros de nuestro sistema y a cada uno le asignaremos un peso de forma que el producto vectorial de los parámetros \mathbf{x} con los pesos \mathbf{w} a través de una función sign nos pueda clasificar nuestros datos:

$$y(x) = \text{sgn} \left(\sum_{i=1}^{N_v} x_i w_i - \alpha \right) \leftarrow \text{sgn}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

El parámetro α se trata del criterio de activación de la neurona. El perceptrón, buscará crear una separación en los datos marcada por la solución: $\mathbf{x} \cdot \mathbf{w} = 0$ que será una barrera de decisión para clasificarlos.

Los procesos de entrenamiento como hemos visto en difusión se obtienen de estimar el error de nuestro modelo y llevar a cabo el gradiente en función de sus parámetros. Este es un modelo lineal muy sencillo donde los únicos puntos donde no se anule el gradiente de nuestro error será en los puntos mal clasificados. Por lo tanto podemos entender el error como la distancia entre los puntos mal clasificados y la barrera de decisión:

$$\mathbf{E}(\mathbf{x}^{(m)}) = \tilde{\mathbf{y}}^{(m)} \frac{\mathbf{x}^{(m)} \cdot \mathbf{w}}{\|\mathbf{w}\|}$$

Siguiendo los razonamientos del artículo y de la misma forma que hemos usado en difusión podemos deshacernos de factores multiplicativos y tomar el gradiente respecto a los parámetros \mathbf{w} para llegar a:

$$\nabla_{\mathbf{w}} \mathbf{E}(\mathbf{x}^{(m)}) = \tilde{\mathbf{y}}^{(m)} \mathbf{x}^{(m)}$$

Cantidad conforme a la que podemos clasificar los pesos siguiendo:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \gamma \mathbf{y}^{(m)} \mathbf{x}^{(m)}$$

Para estudiar este tipo de algoritmos se ha llevado a cabo un ejemplo de clasificación binaria con la base de datos de MNIST. Esta base de datos recoge muestras de cifras del 0 al 9 manuscritas en un formato de 28x28 píxeles en escala de grises tomando valores entre 0 y 255. La forma en la que se aplicará el modelo será atribuyendo a los píxeles el carácter de parámetros \mathbf{x} y se entrenará con el procedimiento descrito hasta minimizar el número de fallos.

Como resultado obtenemos que para 12665 muestras de entrenamiento se obtiene convergencia en aproximadamente menos de 20 iteraciones del dataset.

Para un training set de 2115 muestras no vistas antes por nuestro modelo obtenemos un único error.

En cuanto a la aplicabilidad de un modelo lineal al estudio principal de difusión vemos que si aplicamos un modelo lineal como el propuesto al proceso reverse lo que esperamos que ocurra es que la distribución $\mathbf{X}^{(T)}$ gaussiana (al ser el ruido desde el cual empezamos la fase generativa del modelo) esperamos que se mantenga gaussiana ante las transformaciones lineales de los parámetros en el proceso reverse [3]