Universidad de los Andes Colombia | Facultad de Ciencias Departamento de Física

Diffusion Probabilistic Models

Proyecto Teórico-Computacional

por

David Leonardo Almanza Márquez

Departamento de Física, Universidad de los Andes
Bogotá D.C, Colombia

Supervisor
Gabriel Téllez

December 2023

Contents

Resumen

En el ámbito del aprendizaje automático los modelos probabilísticos de difusión han surgido como una destacada categoría de modelos generativos. Su objetivo principal es aprender un proceso de difusión que describa la distribución de probabilidad de un conjunto de datos dado. Este enfoque se compone principalmente de dos elementos clave: forward process y backward process. Estos modelos se aplican con éxito a diversas tareas, como el de-noising, el inpainting y la generación de imágenes. Estos modelos han demostrado su versatilidad al generar datos del mundo real, destacando ejemplos notables como generadores de imágenes condicionados por texto, como DALL-E y Stable Diffusion.

La esencia de los modelos generativos basados en difusión encuentra sus raíces en la física estadística, donde la difusión se modela mediante la descripción de la propagación browniana de partículas a través de un sistema físico. Inspirados en estos conceptos, los Diffusion Generative Models toman la noción de difusión como un proceso central. Al comprender cómo las imágenes se difunden en un espacio abstracto, estos modelos capturan patrones inherentes y generan datos sintéticos que reflejan la estructura subyacente de conjuntos de datos reales. Esta conexión con la física estadística proporciona una base teórica sólida para que personas con un cierto conocimiento de física puedan comprender cómo funciona este y otros métodos de Machine Learning, campo que ha tenido un gran desarrollo a lo largo de la última década.

1 | Introducción

La noción revolucionaria de *Diffusion Probabilistic Models* (DPM) se originó en el año 2015, gracias al trabajo pionero de Jascha Sohl-Dickstein en su influyente artículo "Deep Unsupervised Learning using Nonequilibrium Thermodynamics"[1]. En este documento seminal, Sohl-Dickstein introduce un enfoque innovador para el aprendizaje no supervisado de distribuciones de probabilidad a través de la aplicación de principios de la termodinámica fuera del equilibrio. El fundamento de los DPM reside en la construcción de cadenas de Markov que transforman una distribución de probabilidad simple en la distribución de probabilidad deseada para los datos mediante pequeñas perturbaciones. Este concepto proporciona una perspectiva única y poderosa para modelar la complejidad intrínseca de conjuntos de datos, allanando el camino para un nuevo paradigma en la generación y comprensión de imágenes. La investigación posterior, como el trabajo titulado "Denoising Diffusion Probabilistic Models"[2], ha ampliado y refinado estos principios, consolidando a los DPM como una herramienta crucial en el paisaje del aprendizaje automático y la generación de imágenes.

1.1 Planteamiento del problema

Suponga que usted tiene interés en la generación de imágenes de una categoría específica, le surge la necesidad de instruir al ordenador para crear dichas imágenes de manera autónoma. Este desafío va más allá de simplemente utilizar programas de diseño convencionales; implica capacitar al sistema para entender y seguir la distribución de probabilidad asociada con las imágenes de interés.

En el caso específico de imágenes de gatos, se plantea la pregunta crucial: ¿Cómo abordar esta tarea de enseñar al ordenador la compleja distribución de probabilidad que caracteriza a estas imágenes? Aunque poseemos conocimiento visual de cómo se presenta una imagen de un gato, carecemos de información precisa sobre la estructura subyacente de la distribución de probabilidad asociada con estas imágenes.

La dificultad radica en la complejidad inherente de esta distribución de probabilidad. La variedad de características, poses y entornos en los que se pueden presentar los gatos contribuye a la falta de claridad en cuanto a la formulación exacta de dicha distribución. Se destaca entonces el hecho de que, para lograr enseñar al ordenador sobre esta distribución de probabilidad, solo disponemos de un conjunto finito de instancias de imágenes representativas X. Estas imágenes constituyen nuestra única fuente de información observable para abordar este desafío.

La complejidad inherente a la tarea de modelar la distribución de probabilidad de imágenes de gatos a partir de un conjunto limitado de instancias destaca la necesidad de enfoques sofisticados. En este contexto, surge la relevancia de los Diffusion Generative Models, que exploran la aplicación de procesos de difusión para modelar eficazmente estas distribuciones complejas y generar nuevas instancias realistas. Este proyecto se centra en el análisis detallado de la física subyacente a estos modelos, proporcionando una base teórica sólida para comprender y abordar este desafío particular en el ámbito de la generación de imágenes.

1.1.1 Un primer acercamiento al problema

Tenemos ante nosotros un conjunto de imágenes $X = \{x_0, x_1, ..., x_N\}$, cuya distribución de probabilidad subyacente, llamada $q(x_0)$, es la verdadera distribución que sigue este conjunto de datos. Es decir, cada imagen en nuestro conjunto es una muestra de la distribución q.

$$x_i \sim q(x_0)$$

Dado que carecemos de información directa sobre la forma precisa de $q(x_0)$, solo contamos con el conjunto finito de imágenes X. El objetivo es parametrizar, es decir, hallar una distribución de probabilidad $p_\theta(x_0)$ que se aproxime a la distribución real $q(x_0)$ tanto como nos sea posible. Una vez logrado este proceso, podremos generar nuevas imágenes x a partir de $p_\theta(x_0)$.

Un enfoque inicial para abordar este desafío podría ser presentar cada una de nuestras imágenes a la computadora, enseñándole el contenido de manera directa, con la esperanza de que, después de observar todas las imágenes, la computadora aprenda una $p_\theta(x_0)$ efectiva. Sin embargo, este enfoque resultaría ineficiente debido al hecho de que el aprendizaje directo desde X para obtener una aproximación precisa de $q(x_0)$ es impracticable. La dificultad radica en que $q(x_0)$ posee una estructura subyacente intrincada y de alta dimensionalidad, lo que impide un mapeo directo efectivo mediante este enfoque.

En consecuencia, se hace aún más evidente la necesidad de métodos más avanzados y eficientes. Aquí es donde entran en juego los Diffusion Generative Models, ofreciendo una perspectiva única al abordar la complejidad de $q(x_0)$, al modelarla a través de procesos de difusión. En las secciones subsiguientes, exploraremos en detalle cómo estos modelos aprovechan la física estadística para superar los desafíos inherentes en la generación de imágenes a partir de distribuciones complejas y desconocidas.

1.1.2 Volviendo el problema tratable

Consideremos un escenario basado en la física estadística: imaginemos un vaso lleno de agua, dentro del cual hemos dejado caer una gota de tinta. La Figura 1 ilustra este momento inicial $t = 0$, destacando que las partículas de tinta no se han dispersado de manera uniforme en el vaso. Denotamos la distribución de las posiciones de estas partículas como $q(x_0)$ siendo el subíndice 0 indicativo del tiempo $t = 0$

Figure 1.1: Imagen de un vaso de agua con una gota de tinta dentro. Las partículas de tinta aún no se ha disipado uniformemente en todo el vaso de agua. La tinta está distribuída de una manera interesante en el agua. Llamamos a la distribución de las posiciones de las partículas $q(x_0)$. El subíndice 0 es porque nos encontramos en el tiempo $t = 0$. Créditos a SarahRichterArt por la imágen.

Supongamos ahora el deseo de aproximar la distribución de las posiciones de las partículas en este instante inicial, es decir, $q(x_0)$. No obstante, solo disponemos de un conjunto limitado de posiciones observadas, $x \sim q(x_0)$, lo cual hace que intentar aprender directamente esta distribución sea un desafío formidable. La complejidad de q, junto con la variedad de distribuciones que podrían ajustarse al conjunto de datos y la estructura no trivial de las posiciones (como se evidencia en la Figura 1.1), hacen que este enfoque sea impracticable.

En un paralelo directo con el problema de modelar la distribución de probabilidad de un conjunto de imágenes, se plantea la necesidad de extraer la máxima información posible de cada posición x dada. En este contexto, se propone la siguiente idea: explorar cómo las partículas cuyas posiciones conocemos se dispersan a lo largo del tiempo mediante un proceso de difusión.

Introducimos el *Forward Diffusion Process*, que simula el movimiento browniano de cada partícula x utilizando una cadena de Markov. Este proceso, ejecutado en $t \in [0, T]$, donde T representa el tiempo necesario para que la tinta se distribuya homogéneamente ($q(x_T)$ se encuentra en equilibrio térmico) nos permite conseguir una comprensión de la distribución $q(x_t)$ que siguen las partículas después de t pasos en la cadena de Markov de difusión.

Al introducir una dependencia temporal a las posiciones de las partículas, una dependencia que además podemos simular fácilmente, logramos obtener más información del mismo limitado conjunto X. Parametrizamos entonces una función $p_\theta(x_t)$, ahora dependiente del tiempo, y buscamos minimizar la discrepancia entre las distribuciones $q(x_t)$ y $p_\theta(x_t)$. Este proceso se repite para cada instante t, haciendo que la tarea de aprender sea más manejable.

2 | Marco Teórico

Los modelos de difusión probabilística se distinguen por su estructura de dos etapas. La primera de estas etapas es conocida como el *Forward Diffusion Process*. Durante esta fase, el objetivo primordial es la destrucción gradual y controlada de la estructura de los datos. Para alcanzar este propósito, la estrategia principal consiste en seleccionar una imagen $x \sim q(x_0)$ de nuestro conjunto de datos X, y luego introducir ruido gaussiano en la imagen en cada paso de tiempo t. Este proceso imita un movimiento browniano a través del espacio de probabilidad $q(x_t)$ asociado con la imagen, permitiendo así la posterior parametrización de una función $p_\theta(x_t)$ dependiente del tiempo.

La segunda etapa, asociada a la generación de nuevos datos $x \sim p_\theta(x_0)$, es denominada *Backward Diffusion Process*. Durante esta fase, la premisa es comenzar con una imagen $x \sim q(x_T)$[1]. Empleando el método de Langevin Dynamics Sampling, junto con nuestro entendimiento de la distribución de los datos, la meta es revertir el proceso de difusión. De esta manera, se busca conducir la imagen a una posición en el espacio de probabilidad tal que $x \sim q(x_0)$.

Ambos procesos, Forward y Backward Diffusion, están intrínsecamente ligados a los principios de la física estadística. En el Forward Diffusion Process, la simulación de un movimiento browniano refleja la evolución temporal de las partículas en el espacio de probabilidad. En el Backward Diffusion Process, la reversión del proceso de difusión mediante Langevin Dynamics Sampling está fundamentada en principios estadísticos, permitiendo generar nuevas muestras que sigan la distribución original. Se analizará detalladamente cómo estos métodos se apoyan en la física estadística para proporcionar una base teórica sólida a los Diffusion Probabilistic Models.

A lo largo de las secciones a continuación, se abordarán los conceptos clave, se visitarán las ecuaciones pertinentes y se proporcionarán ejemplos ilustrativos para facilitar la comprensión de la conexión entre estos procesos y la física estadística subyacente.

2.1 Forward Diffusion Process

Partimos de nuestro conjunto de datos X, con el objetivo de aprender la distribución de probabilidad subyacente $q(\boldsymbol{x}_0)$. Definimos entonces el proceso de difusión para un elemento $\boldsymbol{x} \in X$ de la siguiente manera:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}), \tag{2.1}$$

Donde $\mathcal{N}(\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I})$ es una distribución gaussiana con promedio $\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}$ y matriz de covarianza $\beta_t \boldsymbol{I}$.

En el tiempo t, la partícula se encuentra en una posición asociada a una distribución de probabilidad gaussiana. Su promedio $\mu_q(\boldsymbol{x}_t, t)$ se define como $\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}$, mientras que su matriz de covarianzas $\Sigma_q(\boldsymbol{x}_t, t)$ es $\beta_t \boldsymbol{I}$. Aquí β_t representa la cantidad de ruido presente en nuestra imagen en el tiempo t. Nuestra variable de ruído está definida para seguir un cronograma que crece linealmente

[1] representando puro ruido gaussiano, ya que $q(x_T)$ refleja la distribución cuando el proceso de difusión ha llegado a equilibrio

desde $\beta_1 = 10^{-4}$ hasta $\beta_T = 0.02$.

Esta formulación captura la formulación markoviana de la dinámica del proceso de difusión, donde la posición de la partícula en un instante t es una función probabilística de su posición inmediatamente anterior $\boldsymbol{x}_{t-1}$. La incertidumbre en la posición está determinada por la cantidad de ruido β_t, que ajusta la dispersión de la distribución gaussiana asociada. Definimos $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$ y esta formulación resulta en

$$
\begin{aligned}
x_1 &= \sqrt{1-\beta_1}\boldsymbol{x}_0 + \sqrt{\beta_1}\epsilon, \\
x_2 &= \sqrt{1-\beta_2}\sqrt{1-\beta_1}\boldsymbol{x}_0 + \sqrt{\beta_2}\epsilon, \\
&\cdots \\
x_t &= \sqrt{1-\beta_t}\sqrt{1-\beta_{t-1}}\ldots\sqrt{1-\beta_1}\boldsymbol{x}_0 + \sqrt{\beta_t}\epsilon.
\end{aligned}
$$

En el contexto de nuestra imagen, cada paso de tiempo (t) implica la adición de un poco más de ruido gaussiano. La expectativa es que, al alcanzar el tiempo $t = T$, hayamos descompuesto completamente la estructura original de la imagen. En este punto, nos encontraremos con un estado de ruido gaussiano puro, como se ilustra en la Figura 2.1. Este proceso refleja la progresiva dispersión y pérdida de información estructural a medida que avanzamos en el tiempo, modelando así la difusión de las partículas en el espacio de probabilidad.

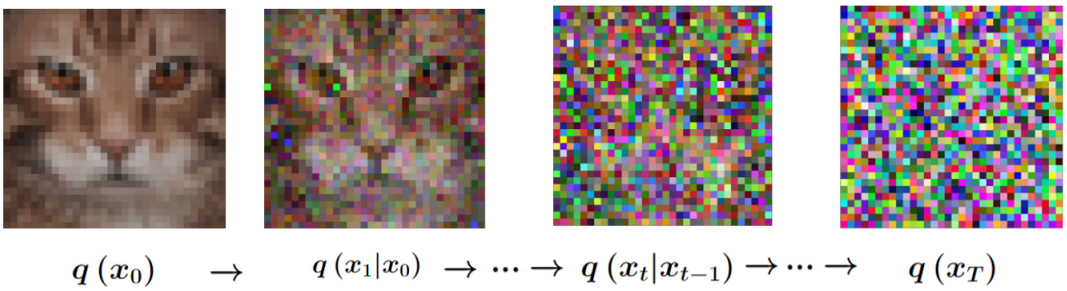

Figure 2.1: Ilustración del Forward Diffusion Process. A medida que avanzamos en el tiempo t, la estructura original de la imagen se descompone gradualmente debido a la adición de ruido gaussiano en cada paso de tiempo. En $t = T$, la imagen resultante es ruido gaussiano puro.

Este proceso de difusión representa una estrategia efectiva para explorar el espacio de probabilidad asociado con la distribución original de la imagen. Al introducir ruido en cada iteración, modelamos el cambio continuo y aleatorio en las posiciones de las partículas, lo que facilita la parametrización de $p_\theta(\boldsymbol{x}_t)$ en función del tiempo.

Para comprender cómo el Forward Diffusion Process se conecta con los principios de la física estadística, consideremos un sistema físico específico: un gas de partículas no interactuantes. Estas partículas experimentan la presencia de un potencial $U(\boldsymbol{x}_0)$. En consecuencia, la distribución de probabilidad de las posiciones de las partículas ($q(x_0)$) se espera que siga la siguiente relación:

$$
q(\boldsymbol{x}_0) \propto e^{-\boldsymbol{U}(\boldsymbol{x}_0)/\beta_t}. \tag{2.2}
$$

En este contexto, anticipamos que todas las partículas tiendan a ubicarse cerca del pozo de potencial. Para simplificar, asumamos que el pozo de potencial es gaussiano, lo que implica que la distribución de probabilidad de las posiciones de las partículas es:

$$
q(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0, \beta_0 \boldsymbol{I}). \tag{2.3}
$$

Nos damos cuenta entonces de que la posición esperada de las partículas es $\boldsymbol{x}_0$, y que nuestro sistema se encuentra a una temperatura $\beta_0 = K_B T_0$[2]. La figura 2.2 nos da una idea de cómo se ve tal sistema.

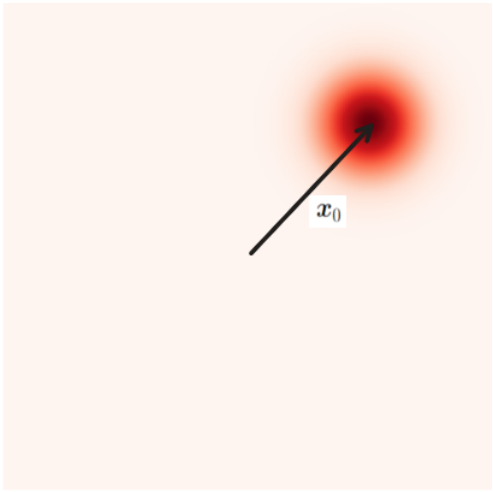

Figure 2.2: Representación esquemática del potencial al que el gas de partículas no interactuantes es sometido. La distribución de probabilidad de las posiciones de las partículas, $q(x_0)$, tiene un promedio $\boldsymbol{x}_0$ y una varianza condicionada por la temperatura actual del sistema β_0.

En este contexto, el Forward Diffusion Process simula el comportamiento de las partículas en el gas cuando aumentamos la temperatura del sistema. Al incrementar la temperatura, regulada por β_t, esperamos observar un cambio en el comportamiento de las partículas. Este cambio se refleja en la capacidad de las partículas para explorar regiones más alejadas del pozo de potencial, ya que ahora disponen de mayor energía térmica. La evolución temporal de la temperatura, controlada por el *schedule* β_t, es fundamental en este proceso.

La idea subyacente es aumentar gradualmente la temperatura hasta que, en el tiempo $t = T$, las partículas del gas se vuelvan (aproximadamente) indiferentes al potencial $\boldsymbol{U}(\boldsymbol{x}_0)$. Este fenómeno se ilustra en la Figura 2.3, donde la distribución de probabilidad de las posiciones de las partículas se expande a medida que la temperatura aumenta, permitiendo que las partículas exploren un rango más amplio de posiciones. La distribución de las partículas corresponde a la cadena de Markov

$$q(\boldsymbol{x}_T|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \tag{2.4}$$

Figure 2.3: Ilustración del aumento de temperatura en el Forward Diffusion Process. A medida que incrementamos la temperatura, controlada por el *schedule* β_t, las partículas del gas ganan energía térmica y exploran regiones más alejadas del pozo de potencial. En $t = T$, las partículas son (aproximadamente) indiferentes al potencial, y se distribuyen de acuerdo a $q(x_T) = \mathcal{N}(0, \boldsymbol{I})$.

[2]En libros de física, se suele usar β para representar la temperatura inversa, sin embargo, para seguir la notación usada normalmente en los papers de estos modelos de machine learning, se usará este nuevo β.

Este enfoque de incrementar gradualmente la temperatura refleja la adaptación del sistema a condiciones más energéticas, lo que conduce a las partículas a un estado donde la influencia del potencial es menos pronunciada. Estas consideraciones son fundamentales para comprender cómo el Forward Diffusion Process modela la exploración y dispersión de las imágenes en el espacio de probabilidad asociado. La progresiva variación en la temperatura permite que las partículas, en este caso, las imágenes, se dispersen y exploren diferentes regiones del espacio de probabilidad.

Es precisamente debido a este proceso que, cuando alcanzamos $t = T$, la imagen resultante corresponde esencialmente a ruido gaussiano. En este punto, la estructura original de la imagen se ha desvanecido completamente, y lo que queda es una representación de ruido gaussiano puro.

Debido a la forma de cadena de Markov que posee el proceso de Forward Diffusion (2.4), podemos introducir las siguientes reparametrizaciones para simplificar la expresión:

$$\alpha_t := 1 - \beta_t, \tag{2.5}$$

donde α_t se interpreta como una especie de medida de la distancia a la temperatura inicial del sistema. Además, definimos:

$$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i, \tag{2.6}$$

y con esto, podemos reescribir la distribución $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ de la siguiente manera:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\overline{\alpha_t}}x_0, (1 - \overline{\alpha_t}\boldsymbol{I}). \tag{2.7}$$

Es importante destacar que, si bien esta reparametrización puede carecer de un significado físico directo para $\overline{\alpha_t}$, nos proporciona la ventaja de poder llegar al tiempo de difusión t directamente desde el tiempo 0. Esta capacidad resulta conveniente para el entrenamiento del modelo, ya que permite presentar a nuestro modelo imágenes aleatorias en momentos t igualmente aleatorios. Esto resulta conveniente porque de esta manera el modelo aprende mejor que si le mostráramos el proceso de difusión de una imagen, en orden, desde el tiempo $t = 0$ hasta el tiempo[3] $t = T$.

2.2 Backward Diffusion Process

Aunque hemos definido el proceso de Forward Diffusion, aún no hemos abordado la manera de aprender la distribución de probabilidad $q(x_t)$ a partir de él. Para abordar este desafío, proponemos parametrizar la función $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$. Es importante señalar que esta distribución está definida para dar pasos desde el tiempo t hasta $t-1$, es decir, retrocede en el tiempo. La idea central es, después de parametrizar con éxito esta distribución de probabilidad, utilizarla para revertir el proceso de difusión de una imagen que sea esencialmente ruido gaussiano puro, es decir, partir de una imagen $x \sim q(x_T)$ y, mediante el proceso de difusión inversa, generar una imagen $x \sim q(x_0)$.

Este enfoque implica retroceder en el tiempo, llevando la imagen desde un estado de ruido gaussiano puro hasta su supuesta forma original. En esta sección, exploraremos en detalle cómo se lleva a cabo esta parametrización y cómo se utiliza para revertir el proceso de difusión, permitiéndonos generar nuevas muestras que sigan la distribución original $x \sim q(x_0)$.

Dada la forma funcional en la que ocurre el proceso de difusión (mediante gaussianas), definimos la parametrización de $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ de la siguiente manera:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) := \mathcal{N}(\mu_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t, t)). \tag{2.8}$$

[3]Esto es evidenciado en resultados experimentales a la hora de entrenar el modelo.

Es importante notar que parametrizar $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ implica parametrizar tanto el promedio μ_θ como la varianza Σ_θ de una serie de distribuciones gaussianas. En otras palabras, debemos aprender el conjunto de parámetros $\boldsymbol{\theta}^*$ que minimiza la diferencia entre las distribuciones p_θ y q, para cada tiempo t. Esta minimización se realiza respecto a la divergencia de Kullback-Leibler (D_{KL}), una medida de la diferencia entre las dos distribuciones de probabilidad p_θ y q. La función de pérdida a minimizar queda expresada como:

$$\mathbb{E}_q[D_{KL}(q(\boldsymbol{x}_T|\boldsymbol{x}_0)||p(\boldsymbol{x}_T)) + \sum_{t>1} D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)||p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) - \log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)]. \tag{2.9}$$

Esta explicación representa una heurística para entender la función de pérdida. La minimización de esta función nos permite aprender los parámetros $\boldsymbol{\theta}^*$ que describen el proceso de difusión de manera óptima.

La ecuación 2.9 compara $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ contra $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$. Sin embargo, a través del proceso de difusión, lo que tenemos es $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. En este contexto, $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ es una distribución de probabilidad posterior, calculable si condicionamos en $\boldsymbol{x}_0$ [2]. Teniendo esto en cuenta, definimos

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t \boldsymbol{I}) \tag{2.10}$$

$$\text{con } \tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) := \frac{\sqrt{\overline{\alpha}_t}\beta_t}{1-\overline{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}\boldsymbol{x}_t \text{ y } \tilde{\beta}_t := \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t \tag{2.11}$$

Ahora, profundicemos en la parametrización de $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$, para $1 < t \leq T$. Primero, notemos que, gracias al *schedule* que habíamos definido para nuestro ruido β_t en el forward diffusion process, entonces una aproximación conveniente es $\Sigma_\theta(\boldsymbol{x}_t, t) = \beta_t \boldsymbol{I}$[2]. Esto implica que no necesitamos parametrizar la matriz de covarianzas, ya que conocemos su forma de antemano. Teniendo esto en cuenta, podemos reescribir

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\mu_\theta(\boldsymbol{x}_t, t), \beta_t \boldsymbol{I}). \tag{2.12}$$

Por lo que se propone la siguiente función de pérdida para el tiempo $t-1$:

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\beta_t} \|\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) - \mu_\theta(\boldsymbol{x}_t, t)\|^2 \right] \tag{2.13}$$

Entonces, notamos que lo que debemos predecir es $\tilde{\mu}$, el promedio de la distribución posterior $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$.

Retomemos nuestra analogía en términos de física estadística. Habíamos definido que $q(x_{t-1})$ representa la distribución de probabilidad de las posiciones de las partículas x, en el tiempo $t-1$. Reescribamos tal distribución, teniendo en cuenta las nuevas variables que hemos definido:

$$q(\boldsymbol{x}_{t-1}) = \frac{1}{Z_0} e^{-\frac{1}{2\beta_t}(\boldsymbol{x}_{t-1}-\tilde{\mu})^2}, \tag{2.14}$$

donde Z_0 es la constante de normalización. Observamos que la ecuación 2.13 se reduce a una función de pérdida cuadrática entre los gradientes $\nabla_x \log p_\theta(\boldsymbol{x}_{t-1})$ y $\nabla_x \log q(\boldsymbol{x}_{t-1})$.

Estamos buscando los parámetros $\boldsymbol{\theta}^*$ que minimizan la diferencia entre la posición esperada real $\tilde{\mu}$ y la posición esperada de nuestra distribución parametrizada μ_θ (ver figura 2.4). Este proceso se repite para cada tiempo t en la cadena de difusión. Notablemente, esta formulación elimina la necesidad de calcular la constante de normalización Z_0, que puede ser computacionalmente desafiante de determinar debido a la alta dimensionalidad del espacio de probabilidad de las imágenes.

Podemos continuar simplificando la función de pérdida (2.13). Para esto, recordemos que

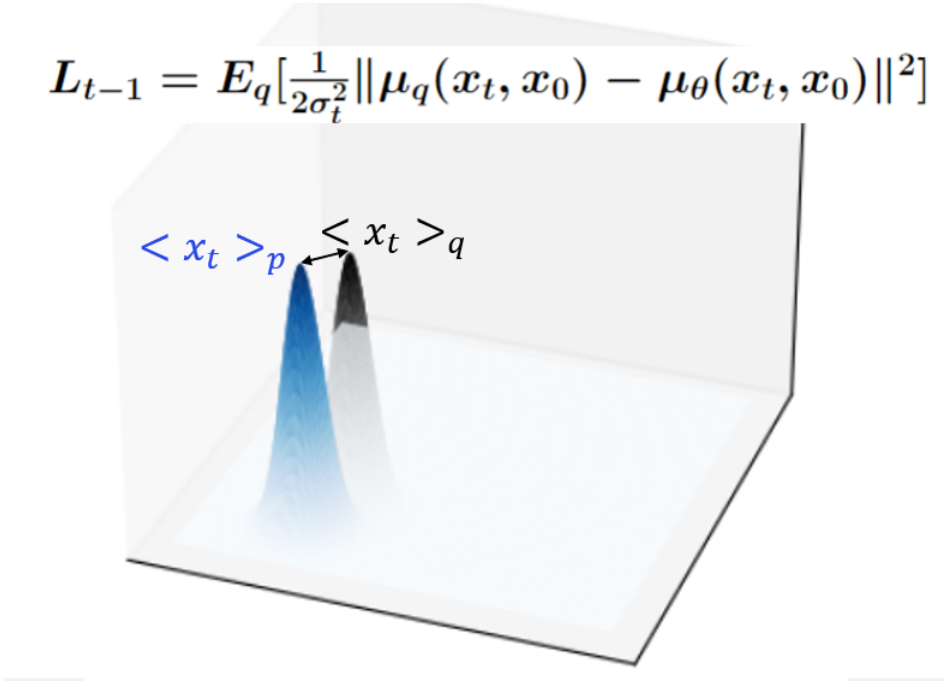

Figure 2.4: Representación gráfica del proceso de optimización. La figura ilustra la minimización de la diferencia entre la posición esperada real $\tilde{\mu}$ y la posición esperada de la distribución parametrizada μ_θ. Este procedimiento se repite para cada paso de tiempo t en la cadena de difusión.

$$\boldsymbol{x}_t(\boldsymbol{x}_0, \epsilon) = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, \text{ con } \epsilon \sim \mathcal{N}(0, \boldsymbol{I}). \tag{2.15}$$

Despejamos x_0 de (2.15), sustituimos en (2.13), y encontramos

$$L_{t-1} = \mathbb{E}_{\boldsymbol{x}_0,\epsilon}\left[\frac{1}{2\beta_t}\left\|\tilde{\mu}_t\left(\boldsymbol{x}_t(\boldsymbol{x}_0,\epsilon), \frac{1}{\sqrt{\overline{\alpha}_t}}(\boldsymbol{x}_t(\boldsymbol{x}_0,\epsilon) - \sqrt{1-\overline{\alpha}_t}\epsilon)\right) - \mu_\theta(\boldsymbol{x}_t(\boldsymbol{x}_0,\epsilon), t)\right\|^2\right]. \tag{2.16}$$

Usamos la definición de $\tilde{\mu}_t$ (2.11), y simplificamos

$$L_{t-1} = \mathbb{E}_{\boldsymbol{x}_0,\epsilon}\left[\frac{1}{2\beta_t}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t(\boldsymbol{x}_0,\epsilon) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon\right) - \mu_\theta(\boldsymbol{x}_t(\boldsymbol{x}_0,\epsilon), t)\right\|^2\right], \tag{2.17}$$

mostramos entonces que μ_θ debe predecir $\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t(\boldsymbol{x}_0,\epsilon) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon\right)$. Lo único que desconocemos allí es ϵ. Escogemos entonces la parametrización para μ_θ:

$$\mu_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t(\boldsymbol{x}_0,\epsilon) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_t, t)\right), \tag{2.18}$$

y notamos que estamos intentando predecir ϵ. Esto lo haremos parametrizando la función ϵ_θ. La fórmula de pérdida final es

$$L_{t-1} = \mathbb{E}_{\boldsymbol{x}_0,\epsilon}\left[\frac{\beta_t}{2\alpha_t(1-\overline{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]. \tag{2.19}$$

Esta ecuación nos dice que estamos intentando predecir el ruido que se le ha agregado a la imagen $\boldsymbol{x}_t$, dado que han pasado t pasos en la cadena de difusión. Para cumplir con esta tarea, contamos con $\boldsymbol{x_0}$, la imagen sin ruido. El algoritmo final de entrenamiento es:

2.2.1 Algoritmo de entrenamiento

1. **Repetir hasta converger**

2. Tomar una imagen $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$

3. Muestrear un tiempo aleatorio de $t \sim \text{Uniform}(\{1, 2, ..., T\})$

4. Muestrear $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

5. Tomar descenso del gradiente en $\nabla_\theta \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2$

Hasta este punto hemos definido el proceso de entrenamiento del modelo, y con ello, hemos encontrado una forma de hallar el conjunto de parámetros $\boldsymbol{\theta}^*$ que aproxima efectivamente a la función parametrizada $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ a la distribución real de las imágenes $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$. En términos de nuestra analogía física, hemos aprendido una distribución p que describe de manera precisa la posición de las partículas en cada instante t.

Este logro es significativo ya que, al intentar comprender la distribución intrincada de las posiciones $q(\boldsymbol{x}_0)$ en el sistema, de la cual solo tenemos información sobre la posición de un conjunto limitado de partículas, decidimos aprender cómo se ve afectada la posición de cada partícula dado que el sistema se encuentra a una temperatura β_t. Repetimos este proceso iterativo hasta obtener una aproximación satisfactoria del conjunto de distribuciones $q(x_t)$.

El siguiente paso implica generar nuevas imágenes que sigan la distribución de probabilidad recién aprendida. Para este propósito, emplearemos un método de muestreo conocido como Langevin Dynamics Sampling, que también se basa en principios de física estadística.

2.3 Generación de nuevas imágenes a través de Langevin

Hasta este momento, hemos alcanzado el logro de derivar un conjunto de distribuciones $p_\theta(\boldsymbol{x}_0)$ que se espera que facilite la generación de nuevas imágenes $\boldsymbol{x}_0$ de acuerdo con la distribución real $q(\boldsymbol{x}_0)$. En esta sección, nos sumergiremos en el estudio del método que emplearemos para materializar este propósito: el *Langevin Dynamics Sampling*. Previo a detallar su aplicación específica en el contexto de los Modelos Probabilísticos de Difusión, exploraremos una contextualización general del método. Posteriormente, examinaremos cómo se integra este enfoque en nuestra investigación, destacando las consideraciones específicas necesarias para el éxito dentro del marco de los modelos de difusión probabilística.

2.3.1 Langevin Dynamics

En física, la ecuación de Langevin es una descripción matemática del movimiento browniano de partículas en un medio, donde se ven afectadas por fuerzas deterministas y estocásticas [3]. La ecuación es:

$$m\frac{d^2x}{dt^2} = -\nabla U(x) - \gamma m\frac{dx}{dt} + \sqrt{2m\gamma k_B T}\eta(t), \tag{2.20}$$

aquí, m es la masa de la partícula, $U(x)$ es el potencial al que está sujeta la partícula (generando una fuerza $-\nabla U(x)$), γ es la constante de amortiguación que modela la resistencia del medio (como un solvente) al movimiento de la partícula, y $\eta(t)$ es una fuerza estocástica que representa el impacto de las colisiones aleatorias con otras partículas del medio.

La presencia de la fuerza estocástica $\eta(t)$ es característica del movimiento browniano y está relacionada con la temperatura T del sistema a través de la constante de Boltzmann k_B. Esta fuerza aleatoria refleja el impacto térmico y las fluctuaciones en el entorno de la partícula.

En el caso de un sistema sobreamortiguado con $m = 1$ y $\gamma = 1$, la aceleración de la partícula es despreciable, por lo que la ecuación de Langevin se simplifica a una ecuación diferencial de primer orden:

$$\frac{dx}{dt} = -\nabla U(x) + \sqrt{2k_B T}\eta(t). \tag{2.21}$$

2.3.2 Langevin Dynamics Sampling

En el contexto de la generación de imágenes, generar nuevas muestras implica revertir el proceso de difusión para una imagen x_T que corresponda a ruido gaussiano puro. Hemos hallado el conjunto de distribuciones $p_\theta(x_t)$, que además, conocemos corresponde a distribuciones gaussianas. Por la forma gaussiana de las distribuciones, notamos entonces que de alguna manera, lo que en el contexto de ML llaman el *score function* de la distribución de probabilidad $\nabla \log p_\theta(x_t)$ se asemeja a lo que en la sección anterior llamábamos la fuerza determinista $-\nabla U(x)$. En la figura 2.5 se muestra primero una distribución arbitraria $p(x)$, y luego se muestra el score function de tal función. Nótese que la gráfica del score function es un campo vectorial que indica la dirección que debe seguirse para llegar a los máximos de la distribución $p(x)$

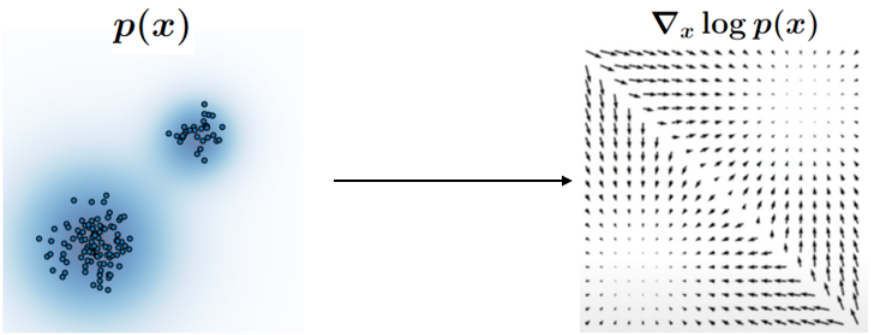

Figure 2.5: se presenta primero una distribución arbitraria $p(x)$, seguida por el score function de esta función. Es crucial notar que la gráfica del score function constituye un campo vectorial que indica la dirección que se debe seguir para alcanzar los máximos de la distribución $p(x)$

La pregunta que surge al considerar la ecuación de Langevin (2.21) es: ¿Por qué no seguir simplemente el campo vectorial sugerido por la *score function*? A continuación, exploraremos la razón por la cual es necesario introducir una fuerza estocástica en el proceso de muestreo.

Como se ha repetido en varias ocasiones, el conjunto de distribuciones reales $q(x_t)$ es bastante complejo. Aunque parametricemos $p_\theta(x_t)$ y logremos que las dos distribuciones se asemejen lo máximo posible al converger el algoritmo de entrenamiento, es normal que p_θ tenga algunas imperfecciones.

En la figura 2.6, se muestra a la izquierda una representación de la distribución real de los datos p_{Data}, y a la derecha la distribución parametrizada lograda, p_θ. Este desajuste puede llevar a problemas durante la generación de datos, ya que una imagen en proceso de generación podría converger a uno de estos mínimos asociados al error de la estimación.

Surge entonces uno de los beneficios de utilizar la ecuación de Langevin: en caso de que la imagen en proceso de generación converja a uno de estos mínimos **locales**, las agitaciones proporcionadas por la fuerza estocástica permitirán a la imagen salir del mínimo, el cual esperamos que no sea muy profundo. De esta manera, la imagen podrá continuar su trayectoria hasta llegar y converger en un mínimo global. Es así como al final, tendremos una imagen $x \sim q(x_0)$.

Para comprender el otro beneficio de utilizar el método de *Langevin Sampling*, consideremos nuevamente el proceso de generación de muestras sin la presencia de la fuerza aleatoria. Supongamos, en

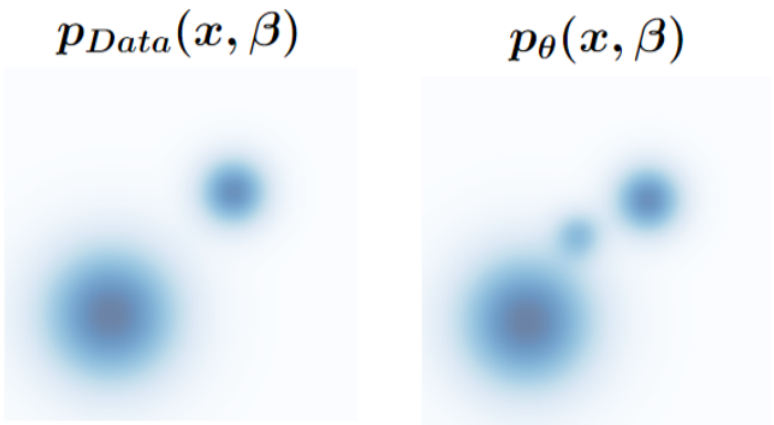

Figure 2.6: A la izquierda, una representación de la distribución real de los datos p_{Data}. A la izquierda, la distribución parametrizada lograda, p_θ.

esta ocasión, que la imagen ha logrado alcanzar un mínimo global, siendo así una imagen $x \sim q(x_0)$. Al finalizar el proceso de muestreo, la imagen estará situada en lo más profundo del pozo. A continuación, deseamos generar una nueva imagen, y al concluir el proceso de generación, nos encontramos con la misma imagen recién generada. Dado que ambas imágenes están ubicadas en lo más profundo del pozo, resultan idénticas. Esta situación no es ideal, ya que el objetivo es generar una nueva y diferente imagen en cada ocasión. No obstante, esto es posible gracias, una vez más, a las fluctuaciones inducidas por la fuerza estocástica. Cuando la imagen logra llegar al pozo, no puede converger exactamente al mismo punto en todas las instancias, lo que posibilita generar una imagen distinta cada vez.

Puntualmente, en el contexto de Diffusion Probabilistic Models, se sigue la versión de la ecuación de Langevin discretizada[2]

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_t, t)\right) + \sqrt{\beta_t}\boldsymbol{z}, \tag{2.22}$$

con $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ la fuerza estocástica. Recordemos además que

$$\mu_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t(\boldsymbol{x}_0, \epsilon) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(\boldsymbol{x}_t, t)\right), \tag{2.23}$$

y que por lo tanto la parte de la fuerza determinista corresponde a un desplazamiento proporcional a la *posición esperada* de las imágenes x en el tiempo siguiente en la cadena de difusión inversa, $t-1$. Este procedimiento iterativo es realizado desde el tiempo $t = T$ hasta $t = 0$. Cuando $t = 0$, tenemos una nueva imagen $x \sim q(x_0)$.

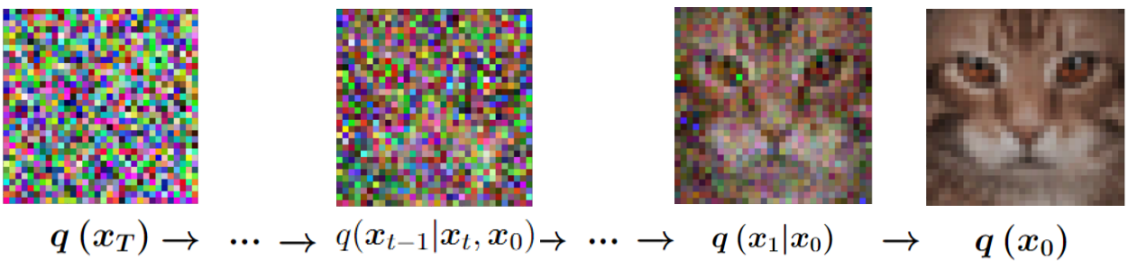

Figure 2.7: Representación del proceso de generación de nuevas imágenes. Se comienza con una imagen de ruido gaussiano x_T, y se sigue la ecuación de Langevin discretizada (2.22) hasta converger en una imagen $x_0 \sim q(x_0)$.

2.3.3 Algoritmo de muestreo

1. Tomar una imagen de ruido gaussiano $x_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
2. **for** $t = T, T-1, ..., 1$ **:**
3. $\quad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
4. $\quad \boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\boldsymbol{x}_t, t) \right) + \sqrt{\beta_t} \boldsymbol{z}$
5. **Fin del for**
6. **retornar** $\boldsymbol{x}_0$.

3 | Implementación y Resultados

Se procedió a la implementación de un Diffusion Probabilistic Model en el lenguaje de programación Python, haciendo uso primordial de la biblioteca PyTorch, una herramienta de vanguardia en el ámbito de la inteligencia artificial. La elección de PyTorch se basó en su flexibilidad y eficacia para la construcción y entrenamiento de modelos de aprendizaje profundo. Esta implementación proporciona una base sólida para explorar y comprender en profundidad las complejidades de los Diffusion Models, permitiendo una experimentación efectiva y una evaluación detallada de los resultados obtenidos. La implementación puede ser vista en *este link*.

3.1 Implementación de prueba

En una fase inicial, se llevó a cabo una implementación de prueba con el propósito de evaluar el correcto funcionamiento del modelo. Durante esta etapa, se entrenó un Diffusion Model con el objetivo de aprender la distribución de probabilidad $q(x_0)$ asociada a una única imagen. En este escenario, $q(x_0)$ se modela como una distribución delta, denotada como $\delta(x - x_0)$, donde x_0 representa la posición específica de la imagen de interés en el espacio de probabilidad. Esta implementación inicial permitió verificar la capacidad del modelo para aprender y representar de manera precisa la distribución de probabilidad de una imagen individual.

En la primera etapa del entrenamiento del modelo (época 0), se observaron los resultados presentados en la figura 3.1. Las gráficas superiores comparan el ruido real añadido a la imagen (izquierda) en un tiempo t arbitrario con el ruido predicho por el modelo sin entrenar (derecha). Además, en la gráfica inferior se contrasta un *flatten* del ruido real $\boldsymbol{\epsilon}$ con el correspondiente a $\boldsymbol{\epsilon}_\theta$, el ruido parametrizado por el modelo. Como se evidencia, antes del proceso de entrenamiento, la predicción no es precisa y muestra la necesidad de ajustar los parámetros del modelo para mejorar la calidad de las predicciones.

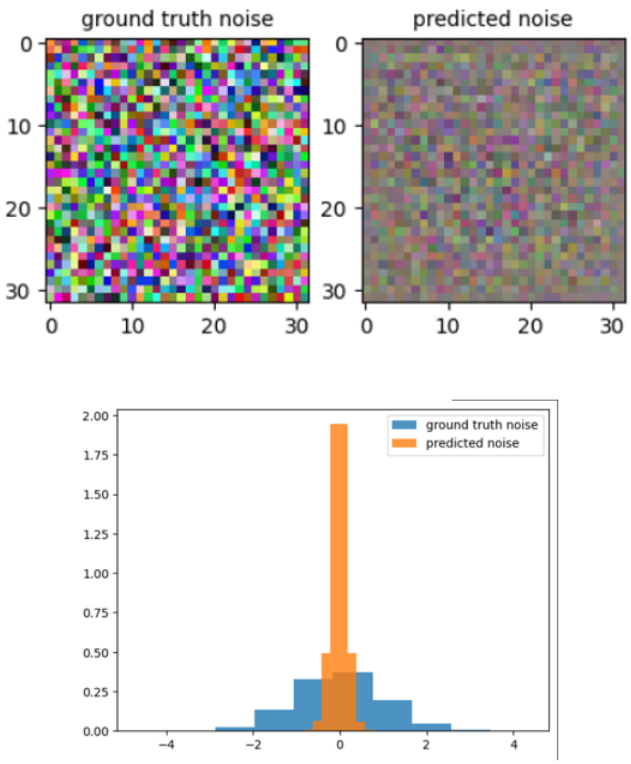

Figure 3.1: Resultados de la época 0 del entrenamiento del modelo Diffusion Probabilistic. Las gráficas superiores comparan el ruido real añadido a la imagen en un tiempo t arbitrario (izquierda) con el ruido predicho por el modelo sin entrenar (derecha). La gráfica inferior contrasta un *flatten* del ruido real $\boldsymbol{\epsilon}$ con el correspondiente a $\boldsymbol{\epsilon}_\theta$, el ruido parametrizado por el modelo antes de cualquier proceso de entrenamiento.

Después de dejar correr el proceso de entrenamiento durante 1600 iteraciones, se observan mejores predicciones por parte del modelo. Esto se puede evidenciar en la figura 3.2, donde se repite el ejercicio de comparación recién realizado. Como se puede observar, ahora el ruido real y el ruido predicho son bastante cercanos. Terminamos entonces el proceso de entrenamiento del modelo, el modelo está listo para generar la imagen sobre la cual se ha *sobreentrenado*.

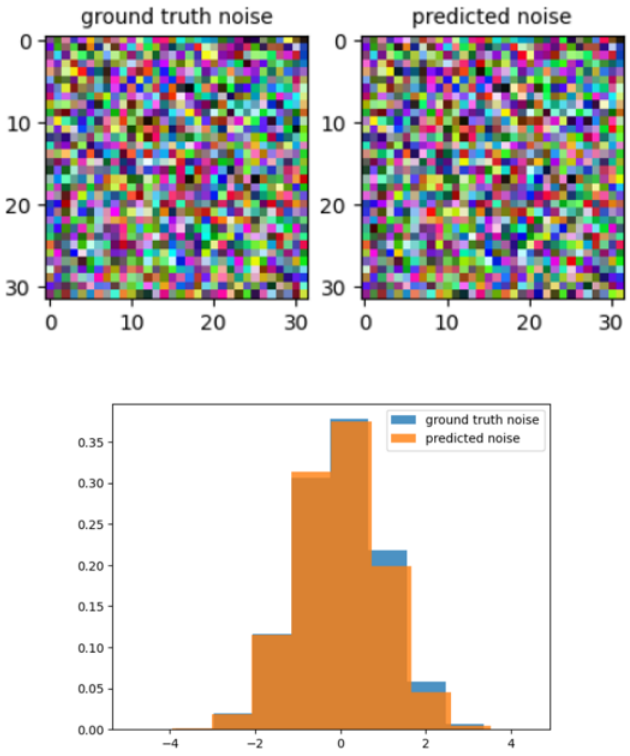

Figure 3.2: Las gráficas superiores muestran la comparación entre el ruido real añadido a la imagen en un tiempo t arbitrario (izquierda) y el ruido predicho por el modelo entrenado (derecha), después de 1600 epochs. En la gráfica inferior, se contrasta un *flatten* del ruido real $\boldsymbol{\epsilon}$ con el correspondiente a $\boldsymbol{\epsilon}_\theta$ después del proceso de entrenamiento. Las predicciones del modelo se acercan notablemente al ruido real, indicando una mejor captura de la distribución de probabilidad.

Finalmente, se llevó a cabo el proceso de generación de acuerdo la forma como se había propuesto en la sección anterior. El resultado final del proceso de generación se muestra en la figura 3.3. Se evidencia que el proceso de muestreo fue exitoso. Estamos listos para ahora sí en verdad poner el modelo a prueba con un conjunto de datos más amplio.

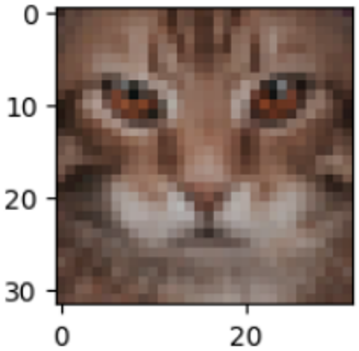

Figure 3.3: Resultado final del proceso de muestreo. Se genera con éxito la imagen sobre la que se entrenó el modelo.

3.2 Implementación con CIFAR-10

La exitosa generación de imágenes a partir del modelo entrenado destaca la capacidad del modelo para aprender y sintetizar información compleja de manera efectiva. Estos resultados alentadores impulsan la evaluación posterior del modelo con el conjunto de datos CIFAR-10, que presenta

un desafío más complejo debido a la diversidad de categorías y características de las imágenes.

Se llevó a cabo un proceso de entrenamiento utilizando el conjunto de datos CIFAR-10, que contiene imágenes de diez categorías distintas. Cada categoría incluye imágenes de aviones, automóviles, pájaros, gatos, ciervos, perros, ranas, caballos, barcos y camiones. Se implementó un nuevo modelo condicionado por categoría para aprender las diez distribuciones de probabilidad correspondientes. Durante el entrenamiento, se utilizó un Batch Size de 256 imágenes y un learning rate de 0.001, repitiendo el proceso durante 100 epochs. La convergencia del entrenamiento se observó a partir del epoch 60, ya que a partir de este punto, la reducción del Loss se estabilizó hasta el final del proceso.

Después del largo proceso de entrenamiento del modelo, se generaron quince muestras para cada una de las categorías. Los resultados se pueden observar en la figura 3.4.

Figure 3.4: Se generaron quince muestras para cada una de las categorías del conjunto de imágenes CIFAR-10.

La evaluación de los resultados demuestra un funcionamiento prometedor en la capacidad del modelo para generar imágenes a partir de la información aprendida durante el entrenamiento. Aunque las imágenes generadas no son completamente reconocibles, se observa la presencia de patrones y características distintivas asociadas a las categorías de las imágenes. La capacidad del modelo para identificar patrones específicos, como los fondos verdes en las categorías de venado y caballo, así como los fondos azules en las categorías de barco y avión, destaca la habilidad del modelo para captar ciertos rasgos relacionados con la categoría de las imágenes.

Es importante tener en cuenta que la calidad de las imágenes generadas podría haberse visto afectada por el reescalado a 32x32 píxeles durante el entrenamiento, una medida tomada para aliviar la carga computacional. Este factor puede haber influido en la claridad y el detalle de las imágenes generadas. A pesar de estas limitaciones, los resultados obtenidos hasta el momento sugieren un potencial significativo del modelo.

En conclusión, aunque los resultados actuales pueden no ser perfectos, representan un avance alentador en la aplicación de modelos de difusión probabilística para la generación de imágenes, especialmente considerando la complejidad y diversidad del conjunto de datos CIFAR-10.

4 | Conclusiones

En este proyecto, nos sumergimos en el mundo de los Diffusion Probabilistic Models (DPM) para la generación de imágenes, explorando su aplicabilidad tanto en un escenario de prueba con una única imagen como en un conjunto de datos más desafiante como CIFAR-10. La esencia del modelo reside en la parametrización del proceso de difusión, que, gracias a la analogía con la física estadística, nos permite entender la generación de imágenes como el movimiento de partículas en un solvente.

En el proceso de entrenamiento, parametrizamos la función de densidad condicional $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ y optimizamos los parámetros para minimizar la diferencia entre las distribuciones reales y parametrizadas. Esta parametrización nos brinda la flexibilidad de generar imágenes a partir de ruido gaussiano, aprovechando el proceso de difusión para simular la evolución temporal.

Introducimos el método de Langevin Dynamics Sampling, una herramienta inspirada en la física estadística, que agrega una componente estocástica al proceso de generación. Esta fuerza estocástica desempeña un papel crucial al evitar que el modelo converja en mínimos locales, asegurando así que las imágenes generadas mantengan la diversidad y evitando la repetición de muestras.

En el contexto de la implementación práctica, comenzamos con una prueba para una única imagen, donde el modelo aprendió exitosamente la distribución de probabilidad asociada. Luego, avanzamos hacia la complejidad de CIFAR-10, entrenando el modelo para aprender las distribuciones condicionales de diez categorías. Aunque los resultados no son perfectos, la capacidad del modelo para reconocer patrones y asociarlos con categorías específicas demuestra su utilidad.

Finalmente, las conclusiones revelan que, a pesar de ciertas limitaciones, como la resolución de las imágenes generadas, el modelo de difusión probabilística presenta un potencial considerable en la generación de imágenes. Este proyecto ha contribuido a nuestra comprensión práctica de los DPM y sienta las bases para futuras investigaciones y mejoras, marcando un emocionante avance en la aplicación de modelos de difusión probabilística en la generación de imágenes.

Bibliography

[1] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. (2015). Deep Unsupervised Learning using Nonequilibrium Thermodynamics. arXiv (Cornell University). http://export.arxiv.org/pdf/1503.03585v8

[2] Ho, J., Jain, A. N., & Abbeel, P. (2020). Denoising diffusion probabilistic models. arXiv (Cornell University). https://arxiv.org/pdf/2006.11239.pdf

[3] Rogel-Salazar, J. (2011). Statistical Mechanics, 3rd edn., by R.K. Pathria and P.D. Beale. Contemporary Physics. https://doi.org/10.1080/00107514.2011.603434

[4] Nick Ali Jahanian. (2022, April 19). MIT 6.S192 - Lecture 22: Diffusion Probabilistic Models, Jascha Sohl-Dickstein [Video]. YouTube. https://www.youtube.com/watch?v=XCUlnHP1TNM

[5] Outlier. (2022, June 6). Diffusion models — Paper explanation — Math explained [Video]. YouTube. https://www.youtube.com/watch?v=HoKDTa5jHvg

[6] Yingzhen Li. (2022, April 22). Learning to generate data by estimating gradients of the data distribution (Yang Song, Stanford) [Video]. YouTube. https://www.youtube.com/watch?v=nv-WTeKRLl0