

# Creación de Patrones de Difracción a través de una GAN Condicional.

Oliver Samuel Pulido Estrada

Universidad Nacional Autónoma de México  
oliver.pe@ciencias.unam.mx

Abril 25, 2025

Resumen.- En éste breve documento se hace la presentación del proyecto final del Diplomado de Inteligencia Artificial y Ciencia de Datos: Métodos fundamentales, PrograLógica, Ciencia de Computación e Ingeniería de Computación, el cuál fue la elección del ejemplo “GAN condicional” de la página [https://keras.io/examples/generative/conditional\\_gan/](https://keras.io/examples/generative/conditional_gan/) modificado para la creación de imágenes sintéticas de Patrones de Difracción.

## 1. Introducción

Los patrones de difracción son manifestaciones de la naturaleza ondulatoria de la luz, que ocurren cuando las ondas de luz interactúan con obstáculos o aberturas. Este fenómeno se puede describir matemáticamente mediante el principio de Huygens-Fresnel y las condiciones de interferencia constructiva y destructiva.

### Principio de Huygens-Fresnel

El principio de Huygens establece que cada punto de un frente de onda actúa como una fuente secundaria de ondas. Cuando la luz pasa por una abertura, las ondas secundarias interfieren entre sí, formando patrones característicos. En el caso de una rendija única, el patrón de difracción está dado por:

$$I(\theta) = I_0 \left( \frac{\sin(\beta)}{\beta} \right)^2$$

donde:

- $I_0$ : Intensidad máxima.
- $\beta = \frac{\pi a \sin \theta}{\lambda}$ : Parámetro de difracción.
- $a$ : Ancho de la rendija.
- $\lambda$ : Longitud de onda de la luz.
- $\theta$ : Ángulo de observación.

### Interferencia en Difracción de Múltiples Rendijas

Cuando la luz pasa a través de múltiples rendijas (por ejemplo, en un sistema de difracción de rendijas paralelas), el patrón se complica debido a la interferencia entre las ondas provenientes de cada rendija. La intensidad está dada por:

$$I(\theta) = I_0 \left( \frac{\sin(\beta)}{\beta} \right)^2 \cos^2 \left( \frac{N \pi d \sin \theta}{\lambda} \right)$$

donde:

- $N$ : Número de rendijas.
- $d$ : Separación entre las rendijas.

## Aplicaciones Prácticas

Los patrones de difracción son esenciales en la cristalografía, que utiliza rayos X para estudiar la estructura de los materiales. Herramientas como DiffraNet analizan automáticamente estos patrones, facilitando la comprensión de la organización atómica en cristales y materiales complejos.

## Generación de Imágenes de Patrones de Difracción mediante GAN Condicionales

Las redes generativas adversarias condicionales (cGAN, por sus siglas en inglés) son una extensión de las GAN que incorporan una condición externa para controlar el proceso de generación de datos. En el contexto de patrones de difracción obtenidos mediante herramientas como DiffraNet, las cGAN pueden ofrecer varias ventajas, entre las que destacan:

### 1. Ampliación de Conjuntos de Datos

En cristalografía serial, los conjuntos de datos suelen ser limitados o estar desbalanceados. Las cGAN pueden generar imágenes sintéticas de patrones de difracción que sean representativas de condiciones experimentales variadas, ampliando así el conjunto de datos disponible para entrenamiento de modelos de análisis.

### 2. Generación Condicional

Las cGAN permiten especificar características del patrón de difracción que se desea generar mediante un vector de condición  $c$ . Por ejemplo, este vector puede incluir parámetros como:

- Tipo de cristal.
- Orientación del cristal.
- Longitud de onda utilizada.
- Condiciones experimentales.

De este modo, las imágenes generadas son coherentes con los parámetros deseados, facilitando simulaciones específicas.

### 3. Validación de Algoritmos

Las imágenes generadas sintéticamente por una cGAN pueden utilizarse para evaluar y validar algoritmos de análisis, como los clasificados mediante DiffraNet. Esto asegura la robustez de dichos algoritmos en diversos escenarios experimentales.

### 4. Reducción de Costos Experimentales

La generación sintética de patrones de difracción puede sustituir, en ciertos casos, la necesidad de realizar experimentos físicos costosos que involucren equipos especializados como generadores de rayos X.

### 5. Exploración de Hipótesis

Las cGAN permiten explorar patrones de difracción en condiciones hipotéticas que serían difíciles o imposibles de recrear experimentalmente. Esto puede conducir a descubrimientos innovadores en la estructura de materiales.

## Aspectos Técnicos

Una cGAN consta de dos redes principales:

- **Generador:** Produce imágenes sintéticas basadas en el ruido  $z$  y la condición  $c$ .
- **Discriminador:** Evalúa si las imágenes son reales o generadas, considerando también la condición  $c$ .

El entrenamiento se realiza mediante la minimización de la función de pérdida adversarial, dada por:

$$\mathcal{L} = \mathbb{E}[\log D(x, c)] + \mathbb{E}[\log(1 - D(G(z, c), c))]$$

donde:

- $x$ : Datos reales.
- $z$ : Vector de ruido.
- $c$ : Vector de condición.
- $G$ : Generador.
- $D$ : Discriminador.

## Programa

A continuación se muestran las características principales de la red modificada:

```
# Generador
def build_generator():
    model = tf.keras.Sequential([
        layers.Dense(8*8*256, use_bias=False, input_shape=(latent_dim,)),
        layers.BatchNormalization(),
        layers.LeakyReLU(),

        layers.Reshape((8, 8, 256)),

        layers.Conv2DTranspose(256, (5, 5), strides=(2, 2), padding='same', use_bias=False),
        layers.BatchNormalization(),
        layers.LeakyReLU(),

        layers.Conv2DTranspose(64, (5, 5), strides=(2, 2), padding='same', use_bias=False),
        layers.BatchNormalization(),
        layers.LeakyReLU(),

        layers.Conv2DTranspose(1, (5, 5), strides=(2, 2), padding='same', use_bias=False, activation='tanh')
    ])
    return model
```

Figura 1: Definición del Generador

```
# Discriminador
def build_discriminator():
    model = tf.keras.Sequential([
        layers.Conv2D(64, (5, 5), strides=(2, 2), padding='same',
            input_shape=[64, 64, 1]),
        layers.LeakyReLU(),
        layers.Dropout(0.3),

        layers.Conv2D(512, (5, 5), strides=(2, 2), padding='same'),
        layers.LeakyReLU(),
        layers.Dropout(0.3),

        layers.Flatten(),
        layers.Dense(1)
    ])
    return model
```

Figura 2: Definición del Discriminador

Intenté modificar todo el código, lamentablemente por que el tiempo se me vino encima no lo conseguí. Al final me perdí en el entrenamiento, la validación y la pérdida y el resultado final fue el siguiente:

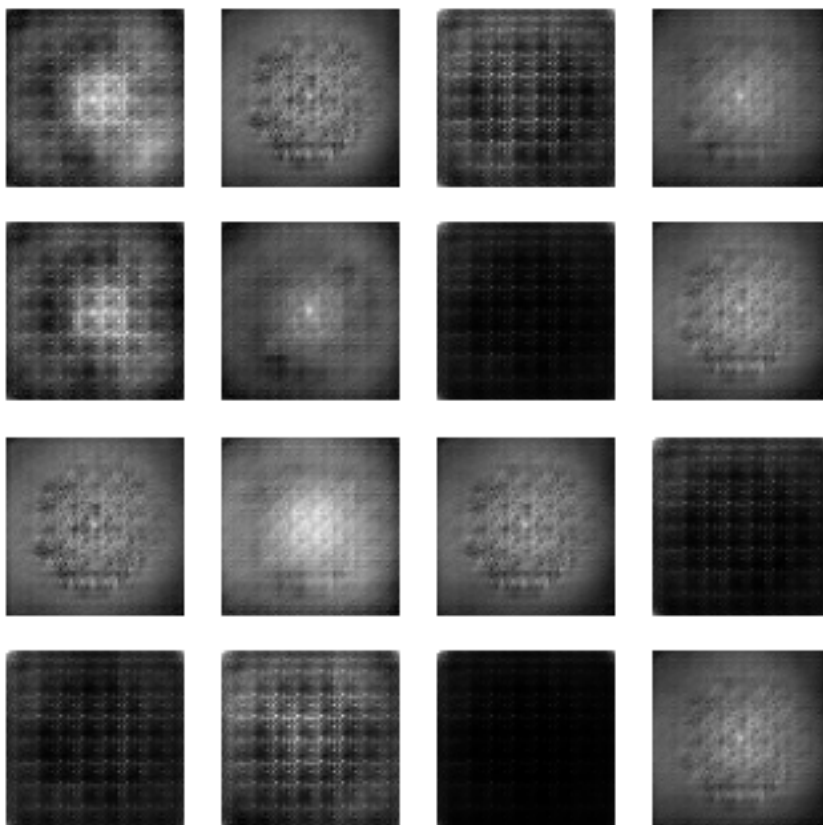


Figura 3: Patrones de Difracción generadas con una cGAN.

## Referencias

- [1] Souza, Artur, Leonardo B. Oliveira, Sabine Hollatz, Matt Feldman, Kunle Olukotun, James M. Holton, Aina E. Cohen y Luigi Nardi: *DeepFreak: Learning Crystallography Diffraction Patterns with Automated Machine Learning*. arXiv preprint arXiv:1904.11834, 2019.
- [2] Luis, Artur: *DiffraNet: Clasificación automática de patrones de difracción en cristalografía serial*. GitHub Pages, 2023. <https://arturluis.github.io/diffranet/>.
- [3] Mirza, Mehdi y Simon Osindero: *Conditional Generative Adversarial Nets*. arXiv preprint arXiv:1411.1784, 2014. <https://arxiv.org/pdf/1411.1784.pdf>.
- [4] Click, Programmer: *Aplicación de GAN en la generación de imágenes*. Programmer Click, 2023. <https://programmerclick.com/article/5214617521/>.