

Guía de Usuario

Sistema de Generación de Prendas de Ropa con DCGAN

Autores:

- Juliana María
- Juan Esteban
- Nikol Tamayo

Curso: Deep Learning Avanzado

Fecha: Diciembre 2025

Introducción

¿Qué es DCGAN Fashion?

Sistema de Inteligencia Artificial Generativa que crea imágenes sintéticas de prendas de ropa utilizando Deep Convolutional GAN entrenado sobre Fashion-MNIST.

Características

- Generación instantánea de hasta 64 prendas
 - 3 experimentos con diferentes configuraciones
 - Interfaz web intuitiva (Streamlit)
 - Control de reproducibilidad mediante semillas
 - Métricas de evaluación profesionales
-

Inicio Rápido

1. Iniciar Aplicación

```
bash

cd IAGENERATIVA_DEEPLARNING
source venv/bin/activate # macOS/Linux
# 0: venv\Scripts\activate # Windows

streamlit run Peña_Atehortua_Tamayo__EA3_GenerativeAI_App.py
```

2. Acceder

Abrir en navegador: `http://localhost:8501`

3. Primera Generación

1. Seleccionar "Experimento 1 – Baseline"
2. Ajustar número de imágenes: 16
3. Clic en " Generar imágenes"

Interfaz de Usuario

Controles Principales

Modelo a utilizar:

- **Experimento 1 – Baseline:** Configuración estándar (10 épocas)
- **Experimento 2 – Más épocas:** Entrenamiento extendido (20 épocas)
- **Experimento 3 – lrD bajo:** LR discriminador reducido

Número de imágenes:

- Rango: 4-64 imágenes
- Recomendado: 16 para balance velocidad/calidad

Semilla aleatoria:

- Rango: 0-9999
- Misma semilla = mismas imágenes (reproducibilidad)

Generar automáticamente:

- Activado: Regenera al cambiar parámetros
- Desactivado: Requiere clic manual

Generación de Imágenes

Comparación de Experimentos

Experimento	Épocas	LR G	LR D	Mejor para
Baseline	10	0.0002	0.0002	Pruebas rápidas
Más épocas	20	0.0002	0.0002	Alta calidad
lrD bajo	20	0.0002	0.0001	Mayor diversidad

Uso de la Semilla

Ejemplos prácticos:

- **Reproducibilidad:** Compartir semilla para obtener mismas imágenes
- **Comparación:** Usar misma semilla en diferentes modelos
- **Exploración:** Cambiar semilla para nuevas variaciones

Entrenamiento de Modelos

Ejecutar Notebook

Google Colab (Recomendado):

1. Subir notebook a Drive
2. Abrir con Colaboratory
3. Runtime → Change runtime type → GPU
4. Runtime → Run all

Jupyter Local:

```
bash

jupyter notebook IAGENERATIVA_DEEPLARNING.ipynb
# Cell → Run All
```

Tiempo de Entrenamiento

- **Con GPU (Colab):** 60-90 minutos
- **Sin GPU (CPU):** 4-6 horas

Archivos Generados

- `modelos/exp1_baseline.pth` (~5-10 MB)
 - `modelos/exp2_mas_epocas.pth` (~5-10 MB)
 - `modelos/exp3_lrD_bajo.pth` (~5-10 MB)
 - Imágenes en carpetas `imagenes_*`
 - Gráficas de EDA y métricas
-

Métricas de Evaluación

FID Score (Fréchet Inception Distance)

- **Mide:** Distancia entre imágenes reales y generadas
- **Rango:** 0 - ∞ (menor es mejor)
- **Bueno:** < 50

Inception Score (IS)

- **Mide:** Calidad y diversidad
- **Rango:** 1 - ∞ (mayor es mejor)
- **Bueno:** > 2.0

Realism Score

- **Mide:** Confianza del clasificador
- **Rango:** 0 - 1 (mayor es mejor)
- **Bueno:** > 0.7

Diversity Score

- **Mide:** Variedad de categorías
 - **Rango:** 0 - 1 (mayor es mejor)
 - **Bueno:** > 0.8
-

Casos de Uso

1. Diseño de Moda

Objetivo: Generar bocetos iniciales para colecciones

Pasos:

1. Usar Experimento 3 (mayor diversidad)
2. Generar 64 imágenes con diferentes semillas
3. Seleccionar diseños interesantes
4. Usar como inspiración

2. E-commerce

Objetivo: Crear mockups para testing A/B

Pasos:

1. Generar variaciones del mismo tipo de prenda
2. Usar diferentes semillas
3. Comparar engagement de usuarios
4. Identificar estilos preferidos

3. Data Augmentation

Objetivo: Aumentar dataset para entrenar clasificadores

Pasos:

1. Generar 1000+ imágenes sintéticas
2. Mezclar con datos reales (80% real, 20% sintético)
3. Entrenar clasificador
4. Evaluar mejora en accuracy

4. Educación

Objetivo: Enseñar conceptos de GANs

Actividad:

1. Mostrar evolución por época
2. Comparar 3 experimentos
3. Explicar impacto de hiperparámetros
4. Permitir experimentación

Preguntas Frecuentes

¿Por qué las imágenes son diferentes cada vez?

Las semillas aleatorias cambian. Usa la misma semilla para reproducir resultados.

¿Puedo generar solo zapatos?

No en esta versión. El modelo genera todas las categorías aleatoriamente.

¿Cuánto tarda el entrenamiento?

Con GPU: 60-90 min. Con CPU: 4-6 horas.

¿Necesito GPU obligatoriamente?

No, pero el entrenamiento es 4-5x más lento en CPU.

¿Dónde se guardan los modelos?

En la carpeta `modelos/` con extensión `.pth`.

¿Puedo compartir los modelos entrenados?

Sí, son archivos `.pth` que puedes compartir.

Error "CUDA out of memory"

Reduce `batch_size` o usa CPU: `device = torch.device("cpu")`.

Las imágenes se ven borrosas

Normal en etapas tempranas. Usa Experimento 2 o 3 para mejor calidad.

Mejores Prácticas

Para Generación

Hacer:

- Experimentar con diferentes semillas
- Comparar los 3 modelos
- Generar lotes pequeños para iterar rápido

Evitar:

- No generar 64 imágenes si solo exploras
- No usar solo Experimento 1 sin comparar

Para Entrenamiento

Hacer:

- Monitorear pérdidas G y D
- Usar GPU para entrenamiento serio
- Documentar cambios en hiperparámetros

Evitar:

- No entrenar más de 30 épocas sin revisar
- No cambiar múltiples parámetros a la vez

Para Análisis

Hacer:

- Combinar métricas cuantitativas y evaluación visual
- Comparar múltiples experimentos
- Documentar resultados

Evitar:

- No confiar solo en una métrica
- No ignorar el contexto de los números

Estructura del Notebook

El notebook contiene las siguientes secciones:

1. Importaciones y configuración
 2. Carga del dataset Fashion-MNIST
 3. Análisis Exploratorio de Datos (EDA)
 4. Definición de Generator y Discriminator
 5. Entrenamiento inicial (Baseline)
 6. Función de entrenamiento con experimentos
 7. Ejecución de 3 experimentos
 8. Evaluación con métricas FID, IS, Realism, Diversity
 9. Comparación y análisis de resultados
-

Soporte

Para problemas técnicos o consultas:

- Revisar documentación del proyecto
- Consultar con los autores del proyecto

Versión: 1.0

Última actualización: Diciembre 2025