
Clase 158 — GANs: DCGAN, Progressive GAN, StyleGAN

Parte: 2 — Deep Learning · Fuente: Géron, cap. 17 § Generative Adversarial Networks + Goodfellow (2014). Duración estimada: 80 min.

Clase 158 — GANs: DCGAN, Progressive GAN, StyleGAN

Parte: 2 — Deep Learning · Fuente: Géron, cap. 17 § Generative Adversarial Networks + Goodfellow (2014). Duración estimada: 80 min.

Objetivo

Construir un GAN (Generative Adversarial Network, Goodfellow 2014) — dos redes en juego adversarial: Generador crea muestras desde noise; Discriminador distingue real de falso. El equilibrio Nash de este juego = generador que genera muestras indistinguibles de la distribución real. Conocer las variantes principales: DCGAN, Progressive GAN, StyleGAN (caras hiperrealistas).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Implementar un DCGAN sencillo con Generator + Discriminator convolucionales sobre MNIST/Fashion.
- Escribir custom training loop alternando G y D updates.
- Diagnosticar 3 problemas clásicos: mode collapse (G produce 1 sola cosa), vanishing G gradient (D demasiado bueno), training inestable.
- Aplicar label smoothing (real labels = 0.9 en lugar de 1.0), noise en D inputs, y WGAN-GP (Wasserstein) para mejor estabilidad.
- Reconocer que GANs perdieron terreno frente a Diffusion (clase 133) en 2022+, pero StyleGAN sigue siendo competitivo para caras.

Temas

- Loss original (min-max): $\min_G \max_D E[\log D(x)] + E[\log(1 - D(G(z)))]$.
- DCGAN: arquitecturas convolucionales (Radford et al. 2015).
- Modos de fallo: mode collapse, D demasiado fuerte/débil.
- WGAN, WGAN-GP, spectral norm.
- Progressive GAN: empezar con 4×4, ir subiendo resolución.
- StyleGAN: AdaIN, style mixing, latent W intermedio.
- Métricas: FID (Fréchet Inception Distance), IS.

Definiciones y características

- Generator: red que toma $z \sim N(0,1) \rightarrow$ genera muestra.
- Discriminator: clasificador binario real/fake.
- Mode collapse: G genera diversidad reducida (1 o pocos modos).
- Vanishing gradient (G): cuando D distingue perfectamente, su gradiente sobre G se desvanece.
- WGAN-GP: cambia la loss a Wasserstein-1 + gradient penalty. Estabilidad superior.
- FID: distancia entre features de InceptionV3 de reales vs falsas; menor = mejor.
- StyleGAN: latent z se mapea a w y se inyecta vía AdaIN en cada capa del generador.

Dataset / recursos

- MNIST / Fashion-MNIST como playground.
- Celeb-A para caras (más serio).
- Librerías: tensorflow, keras.

Ejercicios

1. DCGAN básico: G y D convolucionales. Custom training loop con dos `tf.function` (`train_d`, `train_g`).
2. Diagnóstico: graficar `D_loss` y `G_loss` por step. Si `D_loss` \rightarrow 0, G está perdiendo.
3. Mode collapse: tras N épocas, generar 64 samples. Si todos son la misma cosa \rightarrow collapse.
4. WGAN-GP: implementar gradient penalty en la D loss. Comparar estabilidad vs DCGAN.
5. FID: implementar (o usar `tensorflow_gan.eval.fid`) y reportar FID del modelo.

Homework verificable

DCGAN en Fashion-MNIST:

1. Generator: `Dense(77128)` \rightarrow reshape \rightarrow `Conv2DTranspose` \times 3 hasta 28×28 .
2. Discriminator: `Conv2D` \times 3 \rightarrow `Dense(1)`.
3. Entrenar 50 épocas; generar grid 8×8 .
4. Reportar `D_loss` y `G_loss` finales; visual del grid.

Criterio de aceptación: las muestras generadas son reconocibles como prendas (aunque imperfectas); el training es razonablemente estable (sin colapsar a un solo modo).

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Mode collapse	D muy fuerte, G no puede engañar. Fix: WGA
Training inestable, oscilaciones	LR alto en ambos. Fix: Adam(2e-4, beta_1=0)
<code>D_loss = 0</code>	D resolvió. Fix: hacer D más débil o agreg
Outputs muy borrosos	Underfit de G. Fix: G más expresivo, más é
Outputs de baja resolución bien, alta reso	Sin Progressive Growing. Fix: usar StyleGA

Preguntas frecuentes

¿GAN sigue relevante en 2026?

StyleGAN3 sigue siendo competitivo para caras y generación en general "rápida" (1 forward pass). Difusión gana en calidad para texto-to-image complejo.

¿FID o IS?

FID es mejor (más robusta a artifacts). IS está deprecada.

¿Conditional GAN?

cGAN (Mirza & Osindero 2014): condicionar en label o texto. Permite control sobre qué generar.

¿GAN inversion?

Encontrar z tal que $G(z) \approx \text{imagen_dada}$. Útil para edición. Difícil; StyleGAN tiene buenas implementaciones.

¿Por qué Diffusion ganó?

Más estable de entrenar (sin min-max), mejor calidad de outputs, controllable con text. La explicación más simple: optimiza $\log p(x)$ directamente, GAN optimiza un divergence relacionado pero no la log-likelihood.

Referencias

- Géron, cap. 17 — Generative Adversarial Networks.
- Goodfellow et al. (2014), Generative Adversarial Networks, NeurIPS.
- Radford et al. (2015), DCGAN, ICLR.
- Karras et al. (2018), Progressive Growing of GANs, ICLR.
- Karras et al. (2019, 2020, 2021), StyleGAN1/2/3, NeurIPS/CVPR.
- Gulrajani et al. (2017), WGAN-GP, NeurIPS.

Siguiente clase

Clase 159 — Modelos de difusión (+ Stable Diffusion XL, ControlNet, LCM)

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb