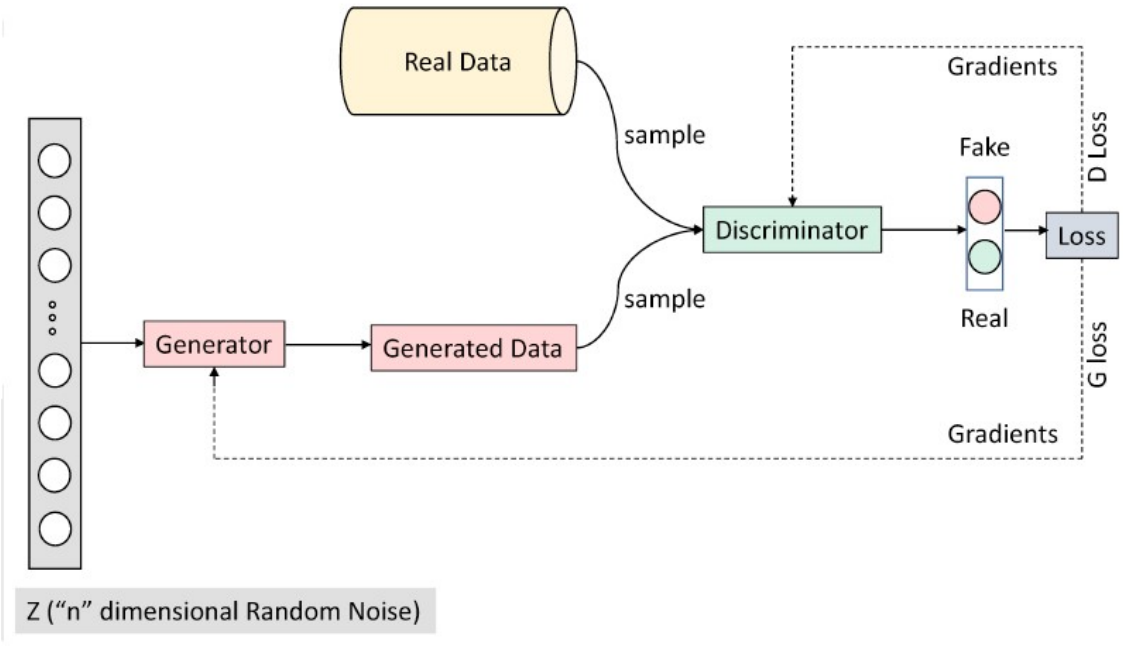
TEORÍA GANs

Las Redes Generativas Adeverstivas (GANs) son una forma de data augmentation (otras formas de data augmentation son las de realizar transformaciones tipo rotaciones o traslaciones de las imágenes iniciales). Su principal objetivo es generar datos (normalmente imágenes) lo más cercanos a la realidad y se suelen usar sobre todo cuando disponemos de un conjunto de datos de tamaño reducido.

Las GANs están compuestas de un generador y un discriminador, que suelen ser redes neurona les y que compiten entre sí durante el entrenamiento. La misión del generador es crear datos p arecidos a los reales mientras que la misión del discriminador es discernir si los datos que le lle gan son reales o sintéticos.

Al generador se le suele pasar como input ruido que sigue una distribución Normal o Uniforme, normalmente.

Si el discriminador funciona mal, la GAN no funcionará correctamente. Esto se debe a que el resto de procesos se alimentan del error producido en el discriminador.



ENTRENAMIENTO DE UNA GAN:

Así, el entrenamiento de la GAN se puede expresar como un problema de MinMax en el que buscamos Maximizar los aciertos del Discriminador sobre los datos reales, y Minimizar los aciertos sobre los datos falsos o sintéticos. De esta forma el discriminador debe acertar el mayor número de casos posibles (se lo queremos poner difícil al generador para que cada vez

lo haga mejor y genere imágenes más realistas) y el generador debe engañar al discriminador y debe generar salidas que se confundan con las reales.

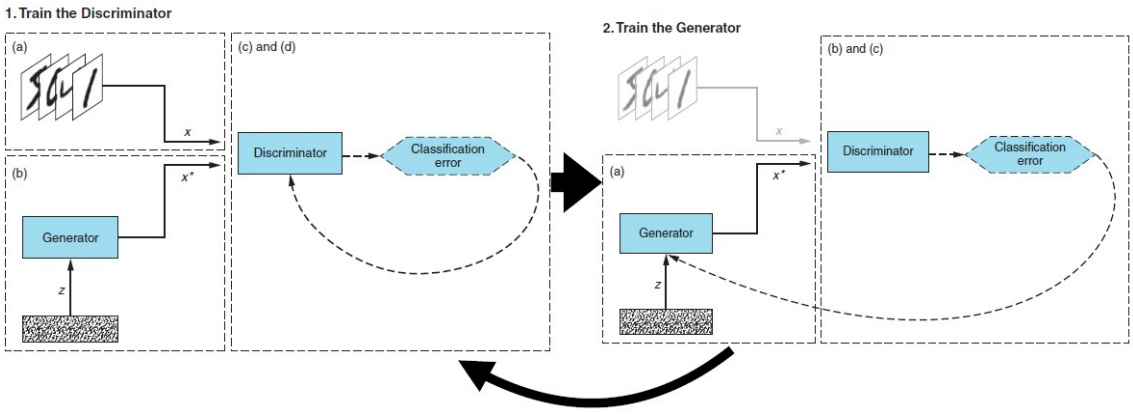
$$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

En la mayoría de aplicaciones, al final de la convergencia se usa el generador solamente. Básicamente la parte que nos interesa es la de generar datos sintéticos.

El proceso de entrenamiento sería el siguiente:

1.- En un primer paso, congelamos el generador y entrenamos el discriminador con datos reales solo. De esta forma, estaríamos enseñando al discriminador lo que es verdad.

2.- En un segundo paso, congelamos los datos reales y solo trabajamos con el generador. Como el discriminador ya sabe lo que es verdad y lo que no, entonces este enseñará al generador cómo crear datos que se parezcan a los reales.



Este proceso se repite para que vayan aprendiendo a la par ambos mecanismos. Cuando el generador sea ya bueno, se dejará de entrenar y solo se utilizará ya el generador. Esto es por lo que hemos dicho antes.

PROBLEMAS DE LAS GANs

Algunos problemas que aparecen al utilizar las GANs son:

1.- **No convergencia y colapso de nodos**: los parámetros del modelo oscilan, se desestabilizan y nunca convergen. Los métodos de descenso por gradiente no funcionan muy bien con los problemas de MinMax. Si el generador es muy bueno, entonces la precisión del discriminador es del 50%, es decir, que la retroalimentación que da es aleatoria y se vuelve menos significativa con el tiempo. Por tanto, *Si la* red generativa continúa entrenando más allá del punto en el que el discriminador está dando una respuesta completamente aleatoria, entonces el generador comienza a entrenar con una respuesta basura por lo que colapsa. Al

colapsar, se produce una variedad limitada de muestras. Hay que dejar que el discriminador vea todo tipo de datos intentando mostrarle la mayor diversidad de casos posible durante su entrenamiento.

2.- **Desaparición del gradiente:** Si el discriminador es demasiado bueno, el entrenamiento del generador puede fallar debido a la desaparición de los gradientes. En efecto, un discriminador óptimo no proporciona suficiente información para que el generador progrese y el entrenamiento se para. Durante el entrenamiento, tanto el discriminador como el generador mejoran iterativamente en sus respectivas tareas hasta que se alcanza un equilibrio en el que el discriminador no puede distinguir las imágenes reales de las sintéticas, prediciendo así D[G(z)] = 0,5 para las imágenes sintéticas.

3.- **Desequilibrio: uno** de los dos modelos aprende más rápido que el otro.

4.- **Inicialización:** Alta sensibilidad a la selección de hiperparámetros. Ahora tenemos dos modelos que deben ajustarse a la vez.

Observando los problemas 1 y 2, recalcamos la idea de que se debe hacer un entrenamiento síncrono entre el generador y el discriminador para que no ocurra que uno es mucho mejor que otro.

EVALUACIÓN DE GANs:

El tema de evaluar el rendimiento de las GANs es algo que es complicado y que existen muchas métricas que se han ido desarrollando, cada una atendiendo a unos criterios u otros, aunque parece que ninguna de ellas termina de convencer dentro de la comunidad. Por tanto, parece inevitable que un humano supervise el generador si se quiere ver la calidad de los datos generados, pero esto se hace imposible se prueba una gran cantidad de hiperparámetros. Las funciones de calidad más extendidas son :

1.- **Inception score:** Tiene en cuenta dos factores: a) la alta predictibilidad de la imagen generada (que sea preciso) y b) que las clases sigan una distribución uniforme.

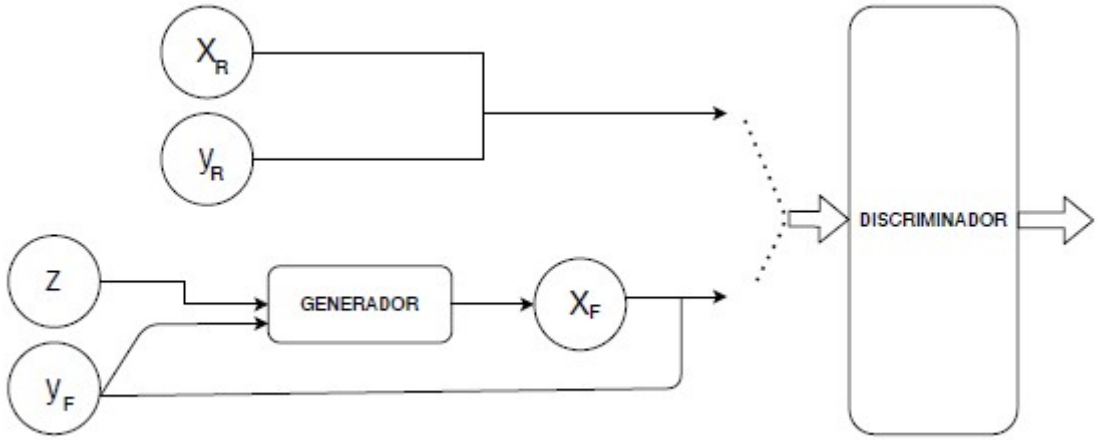
2.- **Fréchet Inception Distance**: se determinan las distancias entre las distribuciones de datos reales y generados usando un modelo preentrenado de Inception V3.

El nombre de Inception procede del clasificador usado para determinar las etiquetas de cada dato generado (Inception, uno de los ganadores de *Imagenet*).

GANs CONDICIONALES:

En las GANs originales no hay forma de determinar *a priori* cómo va a ser la conexión del espacio latente de partida del generador con las variedades de muestras que puede generar. El

discriminador aprende a identificar los pares (dato de entrada más etiqueta) reales que coinciden, mientras que rechaza los pares que no coinciden.



GANs FAMOSAS: VARIACIONES DE LAS GANs:

En principio, el principal cambio de cada variante es la función de coste a minimizar y maximizar, o la arquitectura de las redes neuronales que se utilizan para el generador y el discrminador. El resto de cosas, como la idea de entrenamiento, permanecen iguales.

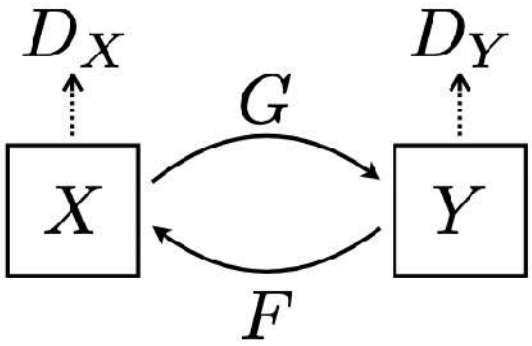
Algunas de las variaciones más conocidas de las GANs son:

1.- **DCGAN (Deep Convolutional GAN):** El Discriminador y el Generador son Redes Neuronales Convolucionales. La entrada del discriminador es una imagen y la salida una probabilidad de que proceda o no de la distribución de los datos reales. La entrada del generador es un vector aleatorio que se extrae de una distribución normal estándar y la salida una imagen. Algunas aplicaciones son la generación de caras sintéticas o de paisajes.

2.- **LSGAN (Leat Squares GAN):** La idea principal de LSGAN es utilizar una función de pérdida que proporcione un gradiente suave y que no se sature en el discriminador D. Las LSGANs son capaces de generar imágenes de mayor calidad que las GANS normales y proporcionan entrenamientos más estables.

3.- **WGAN (Wasserstein GAN):** Utiliza la distancia de Wasserstein en la función de pérdida. Esta distancia mide el ínfimo del promedio entre la cantidad que hay que desplazar del punto x al punto y, por el coste de ese desplazamiento. En particular, mide la similitud entre dos distribuciones. Soluciona el problema del desvanecimiento del gradiente y es bastante estable.

4.- **CYCLEGAN:** Se tienen dos GANs funcionando de forma cíclica, donde el discriminador debe diferenciar si los datos proceden de su dominio o bien del generador que toma los datos del otro dominio; se tendrían entonces dos problemas *minimax* como los definidos en las GANs originales. Permite jugar con datos de diferentes dominios, por ejemplo imágenes y texto.



ENTRENAMIENTO PROGRESIVO DE UNA GAN:

La generación de imágenes de alta resolución es complicada ya que al tener más detalles, hace más fácil que se puedan distinguir las imágenes sintéticas de las reales, y tendríamos así el problema del gradiente comentado anteriormente. Además, a mayor resolución más memoria necesitamos y tendríamos que usar mini batches más pequeños.

El entrenamiento progresivo consiste en ir aumentando la resolución de las imágenes que le pasamos a la GAN (generador+discriminador) poco a poco. Con esto conseguimos que primero se aprenda la estructura general de las imágenes y que luego se vaya fijando en los detalles, en lugar de que la red tenga que aprenderlo todo de forma simultánea. El hecho de aumentar la resolución se traduce en ir aumentando el número de capas ocultas de nuestra red. Normalmente esto acelera el entrenamiento y lo estabiliza (que era uno de los problemas que presentaban las GANs), permitiendo así crear imágenes de gran calidad.

En nuestro caso particular, nosotros simplemente hemos implementado 1 step del entrenamiento progresivo. La idea que hemos seguido ha sido entrenar la GAN con imágenes de 28x28, guardarnos los pesos y después transmitirlos a las capas similares de la GAN que será entrenada con imágenes de resolución 56x56. De esta forma, no partimos de cero, sino que ya hemos aprendido ciertas características más generales.

BENEFICIOS DEL ENTRENAMIENTO PROGRESIVO:

Al ir aumentando poco a poco la resolución, estamos haciendo preguntas mucho más simples comparadas con el objetivo final de encontrar un mapeo del espacio latente al espacio de nuestras imágenes. Además, el otro beneficio principal es que se reduce mucho el tiempo de entrenamiento porque la mayoría de iteraciones se realizan con resoluciones de poco tamaño.

Algunas de las mejoras que hemos podido observar del uso del entrenamiento progresivo son:

1.- Una reducción en la cantidad de épocas necesarias para entrenar la red completa. Al partir de capas preentrenadas, los resultados de más alta resolución no necesitan empezar desde cero, tienen un punto de partida y eso acelera el proceso.

2.- Un aumento de la convergencia y la resolución posible. Una alta resolución no sería alcanzable si no fuera por métodos de entrenamiento progresivo, la red simplemente no sabría por donde empezar de la enorme cantidad de caminos disponibles. En nuestro caso, una imagen de 56x56 no era posible con nuestros recursos (hardware), en un solo entrenamiento, pero al segmentarlo en dos etapas, sí se logró la convergencia.