Métricas para la evaluación de GANs

Introducción

- Evaluación de GANs
- Métricas cuantitativas y métricas cualitativas
- Resumen general

Evaluación de GANs

- Desde su introducción en 2014, han existido sustanciales avances en la teoría de las GANs.
- Se han desarrollado múltiples modelos y variantes, adaptadas a diferentes ámbitos, objetivos de investigación y problemas específicos.
- No ha existido tanto avance en el desarrollo de métricas y frameworks integrados para la evaluación de GANs (y otros modelos generativos).

Evaluación de modelos generativos

- Evaluar el desempeño de modelos basados en aprendizaje supervisado: “simple”, dado que el objetivo está claramente formulado y se cuentan con datos concretos para usar como base de la evaluación (ground truth).
- Para los modelos generativos es la evaluación es más compleja:
  - El objetivo es evaluar el realismo de los datos generados.
  - El dataset de datos reales es solamente una (en general pobre) aproximación del conjunto de datos realistas

Evaluación de GANs

- Existen dos tipos de modelos generativos:
  - Explícitos: asumen conocida la función de probabilidad del modelo
  - Implícitos: utilizan mecanismos de muestreo para la generación de datos.
- A diferencia de los modelos explícitos, las GANs no conocen la distribución de los datos reales, sino que tratan de estimarla a través de una distribución paramétrica.

- La gran variedad de criterios probabilísticos y la ausencia de métricas perceptualmente significativas para evaluar la similitud de datos e imágenes, causan que la evaluación de GANs sea notoriamente difícil.

Evaluación de GANs

- Se han propuesto dos enfoques principales para la evaluación de GANs.
  - Evaluación cuantitativa de los modelos, a partir de métricas específicas.
  - Evaluación cualitativa de los modelos: análisis por expertos, estudios de casos de uso, análisis de aspectos internos de los modelos.
- Ambos enfoques tienen fortalezas y limitaciones.
  - Las medidas cuantitativas no son subjetivas, pueden no corresponder directamente a cómo los humanos perciben y juzgan los datos generados.
  - El criterio de “engañar a un humano” en la tarea de distinguir los datos generados de las reales podría ser la prueba definitiva, pero este criterio puede favorecer modelos que se concentran en secciones limitadas de los datos (sobreajuste o memorización), que tienen baja diversidad o sufren de colapso de modo.

Métricas de evaluación de GANs: clasificación

- Métricas cuantitativas, cualitativas y mixtas.
- Univaluadas o (unas pocas) multivaluadas.

Métricas de evaluación de GANs: propiedades deseables

"Metamedidas" para evaluar y comparar las métricas de evaluación de GANs

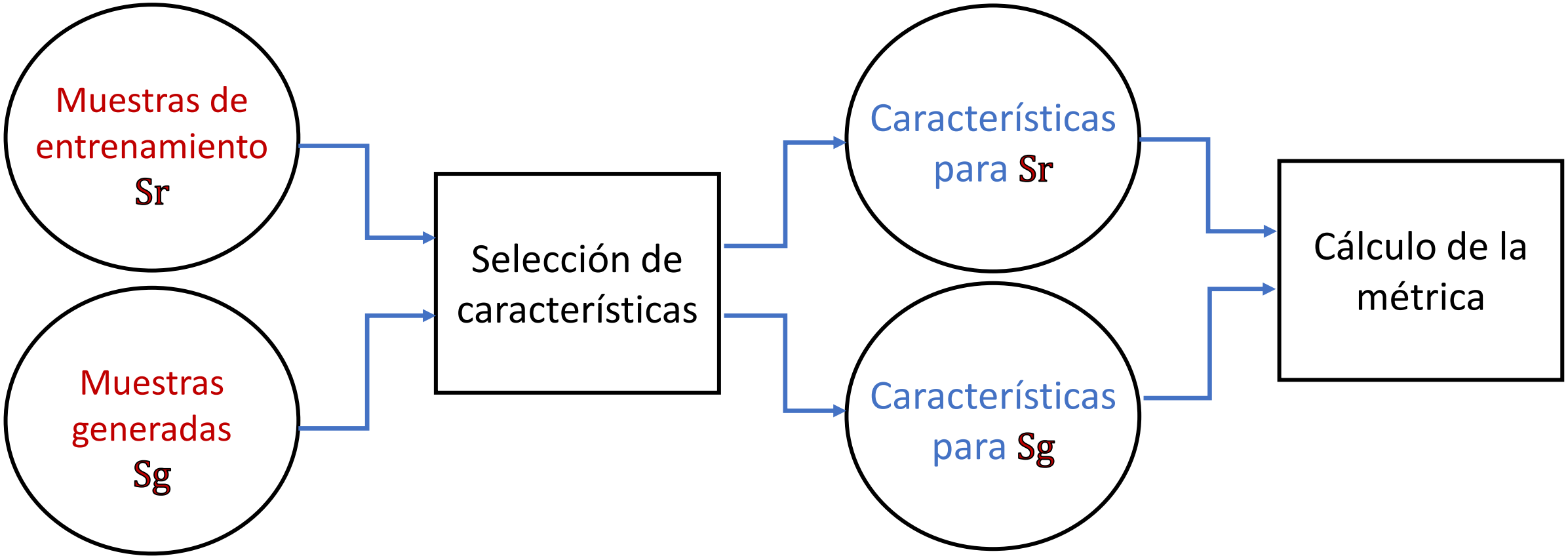
1. Favorecer modelos que generen muestras de alta fidelidad (discriminabilidad: capacidad para distinguir las muestras generadas de las reales)
2. Favorecer modelos que generan muestras diversas (diversidad: modelos robustos al sobreajuste, colapso de modo y gradiente nulo) y que la métrica pueda penalizar modelos triviales como las GANs "que memorizan"
3. Favorecer modelos con espacios latentes desenredados (mapear los factores del espacio latente a una característica del modelo generativo) y con continuidad espacial (muestreo controlable)
4. Tener cotas bien definidas: inferior, superior y probabilísticas (estimabilidad)

Métricas de evaluación de GANs: propiedades deseables

5. Sea robusta a las distorsiones y transformaciones de los datos de entrada que no cambian los significados semánticos (invariabilidad semántica). Por ejemplo, el score de un generador entrenado en el conjunto de datos de caras de personas famosas (CelebA) no debería cambiar significativamente si las caras generadas se desplazan unos pocos píxeles o se rotan en un ángulo pequeño
6. Ser coherente con los juicios perceptivos humanos y las clasificaciones humanas de modelos (coherencia)
7. Tener baja complejidad computacional y muestral (eficiencia)

Métricas cuantitativas

- Métricas basadas en valores calculados sobre las distribuciones de los datos generados y de los datos reales



Métricas cuantitativas

- Dos tipos de métricas cuantitativas
- Métricas “agnósticas del modelo”: usan el generador como caja negra para muestrear salidas y no requieren una estimación de densidad del modelo construido.
- Métricas que asumen conocimiento del modelo: requieren estimar una distribución de probabilidad a partir de muestras de las salidas.

Métricas cuantitativas: average log-likelihood

- Los métodos de estimación de densidad de kernel (Kernel Density Estimation, KDE) estiman la función de densidad de una distribución a partir de muestras.
- La divergencia de Jensen Shannon (JSD) se usa para estimar la densidad de GANs, pero algunos autores las han cuestionado (Theis et al. 2015)
- Log-likelihood (o divergencia de Kullback-Leibler, KLD) se ha usado como estándar de facto para entrenar y evaluar modelos generativos.
- Mide la verosimilitud de los datos reales (held out/test) bajo la distribución generada con N muestras $L = \frac{1}{N} \sum_i \log P_{model}(\mathbf{x}_i)$
- Se utiliza el modelo generado para inferir la log-likelihood, asumiendo que un modelo que la maximiza (KLD=0) es capaz de generar muestras perfectas.

Métricas cuantitativas: average log-likelihood

- Probabilidad conjunta de los datos observados como función de los parámetros del modelo
- La verosimilitud es una métrica muy intuitiva, pero tiene desventajas:
  - Los métodos de estimación no son fiables en espacios de alta dimensión y en espacios medianos pueden requerir de un gran número de muestras para estimar razonablemente el verdadero log-likelihood del modelo;
  - La verosimilitud no aporta información concreta sobre la calidad de los datos generados (y viceversa): log-likelihood y la calidad de las muestras están moderadamente no relacionadas.

Métricas cuantitativas: average log-likelihood

- Un modelo puede tener log-likelihood pobre y producir excelentes muestras, o tener muy buen log-likelihood y producir muestras mediocres.
  - Para una mezcla de distribuciones gaussianas cuyas medias se corresponden con los datos de entrenamiento, un modelo puede generar muy buenas muestras, pero tendrá log-likelihood muy pobre.
  - Una combinación de dos modelos, uno bueno pero con una ponderación baja (e.g., 0.01) y uno malo con un peso alto, tendrá valores altos de average log-likelihood, pero generará muestras muy malas.
- Resulta complejo determinar si las GANs simplemente “memorizan” los datos de entrenamiento o si pierden modos importantes de la distribución de datos reales.

Métricas cuantitativas: cobertura

- Probabilidad de que los datos reales están “cubiertos” por la distribución del modelo generado

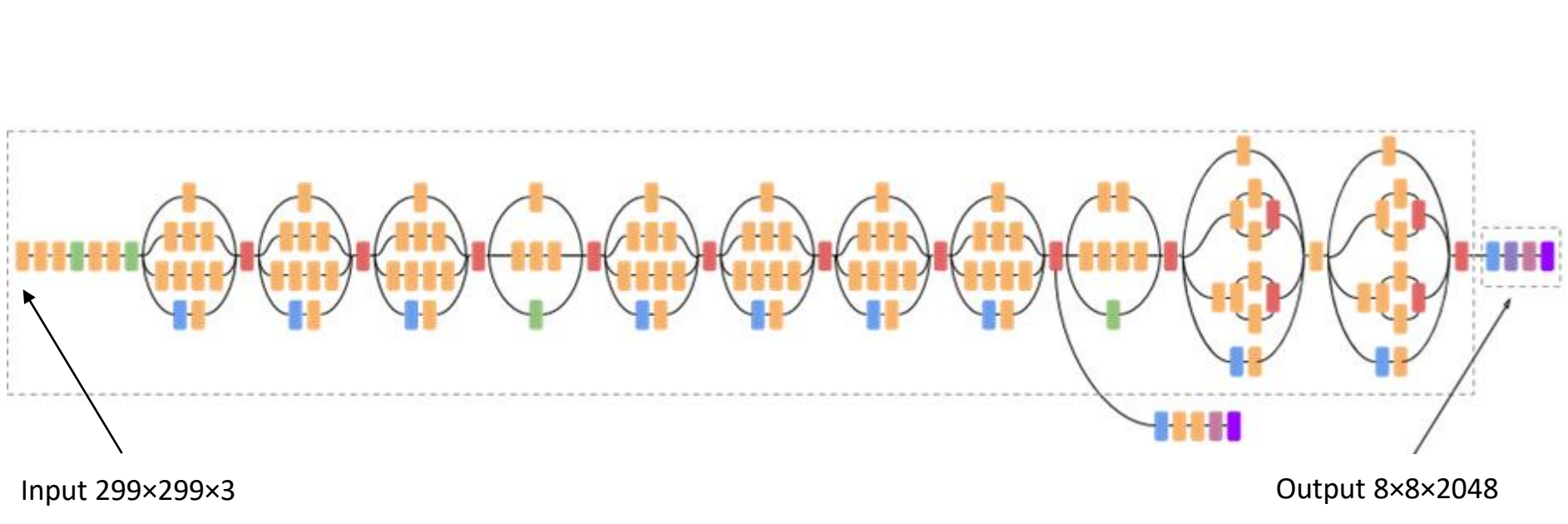
  $C = P_{data}(P_{model} > t)$, con t tal que $P_{model}(P_{model} > t) = 0.95$

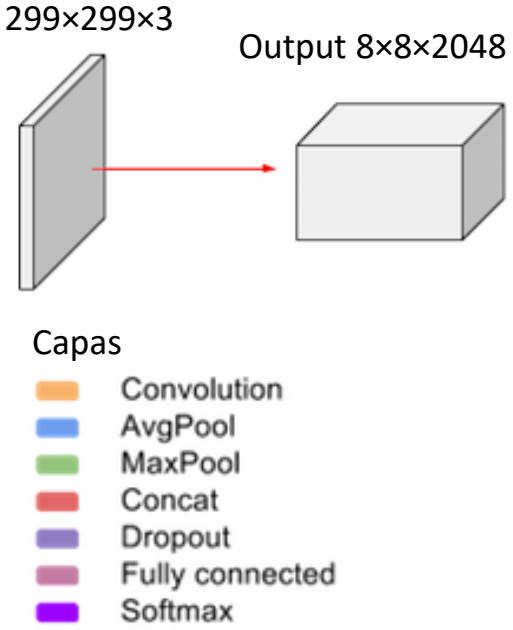
- Para aproximar la densidad de P_{model} se utiliza un método de estimación de densidad de kernel.
- Tolstikhin et al. (2017) sugirieron que es una métrica más interpretable que la verosimilitud, que simplifica la tarea de comparar la eficacia de GANs.

```
model_log_density = kde.score_samples(synthesized_data)
threshold = percentile(model_log_density, 5)
real_data_log_density = kde.score_samples(real_data)
ratio_not_covered = mean(real_data_log_density <= threshold)
C = 1 - ratio_not_covered
```

Métricas cuantitativas: inception score

- Inception Score (IS) es una de las métricas más ampliamente aceptadas para la evaluación de GANs que trabajan con imágenes.
- Utiliza una ANN (Google inception network) preentenada en ImageNet para capturar propiedades deseables de las muestras generadas: altamente clasificables y diversas con respecto a las etiquetas/clases.





Métricas cuantitativas: inception score

- IS trata de formalizar el concepto de realismo de los datos generados mediante dos criterios:
  1. Cada dato generado debe ser reconocible (discriminable), por lo cual la distribución debe estar idealmente dominada por una clase.
  2. La distribución de clases para la muestra debe ser lo más cercana posible a una distribución uniforme (evalúa la diversidad del generador)
- Formalización:
  1. Las distribuciones intra-clase deben tener baja entropía
  2. La entropía de la distribución global debe ser alta

Métricas cuantitativas: inception score

- Formalización:
  1. Las distribuciones des clases de imágenes deben tener baja entropía
  2. La entropía de la distribución global debe ser alta
- Los criterios refieren a la salida del clasificador preentrenado (Inception Net): se calcula la distancia KL entre las distribuciones.
- Un conjunto de imágenes reales que cumple ambos criterios tendrá IS alto
- Un conjunto de imágenes aleatorias o generadas por una GAN que colapsa en una clase tendrá IS bajo.

Métricas cuantitativas: inception score

- IS mide la KLD promedio entre la distribución condicional de etiquetas de las muestras $p(y|\mathbf{x})$ (se espera que tenga baja entropía para muestras de mejor calidad, fácilmente clasificables) y la distribución marginal $p(y)$ obtenida de todas las muestras (se espera que tenga alta entropía si hay alta diversidad, todas las clases están bien representadas en el conjunto de muestras)
- Favorece una baja entropía de $p(y|\mathbf{x})$ pero una gran entropía de $p(y)$.

- IS es razonablemente confiable para filtrar malos resultados

Inception score con numpy

```
def calculate_IS(p_yx, eps=1E-16):
	# calcular p(y)
	p_y = expand_dims(p_yx.mean(axis=0), 0)
	# KLD para cada imagen
	kl_d = p_yx * (log(p_yx + eps) - log(p_y + eps))
	# suma sobre clases
	sum_kl_d = kl_d.sum(axis=1)
	# promedio sobre imágenes
	avg_kl_d = mean(sum_kl_d)
	# calcular IS (remover logaritmos)
	is_score = exp(avg_kl_d)
	return is_score
```

Inception score en Keras

```
# asume imágenes de 299x299x3, con pixels en [0,255]
def calculate_inception_score(images, n_split=10, eps=1E-16):
	# cargar el modelo inception v3
	model = InceptionV3()
	# convertir a punto flotante
	processed = images.astype('float32')
	# preprocesar imágenes para el modelo inception v3
	processed = preprocess_input(processed)
	# predecir la probabilidad de cada clase
	yhat = model.predict(processed)
	# enumerar particiones de imágenes y predicciones
	scores = list()
	n_part = floor(images.shape[0] / n_split)
```

Inception score en Keras

```
for i in range(n_split):
        # recuperar p(y|x)
        ix_start, ix_end = i * n_part, i * n_part + n_part
        p_yx = yhat[ix_start:ix_end]
        # calcular p(y)
        p_y = expand_dims(p_yx.mean(axis=0), 0)
        # calcular KLD (usando log de las probabilidades)
        kl_d = p_yx * (log(p_yx + eps) - log(p_y + eps))
        # suma sobre clases y promedio sobre imágenes
        sum_kl_d = kl_d.sum(axis=1)
        avg_kl_d = mean(sum_kl_d)
        # revertir el logaritmo y almacenar
        is_score = exp(avg_kl_d)
        scores.append(is_score)
# promedio sobre imágenes
is_avg, is_std = mean(scores), std(scores)
return is_avg, is_std
```

Inception score en Keras

- Ejemplo de invocación

```
# generar imágenes (cargar salida del generador)
images = ones((50, 299, 299, 3))
print('loaded', images.shape)
# calcular IS
is_avg, is_std = calculate_inception_score(images)
print('score', is_avg, is_std)
```

- Código disponible en https://colab.research.google.com/drive/19xOYNmiKihNKC4FcsAqOZaDf3gCbBqF-?usp=sharing

Métricas cuantitativas: inception score

- IS tiene una correlación razonable con la calidad y la diversidad de los datos generados
- El valor de IS calculado sobre datos reales es una cota superior para analizar el valor de métrica.
- Desventajas:
- Al igual que log-likelihood, favorece “GANs con memoria” de los casos de entrenamiento: no es capaz de detectar overfitting y puede engañarse generando centros de los modos de los datos. Este hecho es agravado porque no utiliza un holdout/training set para validación.

Métricas cuantitativas: inception score

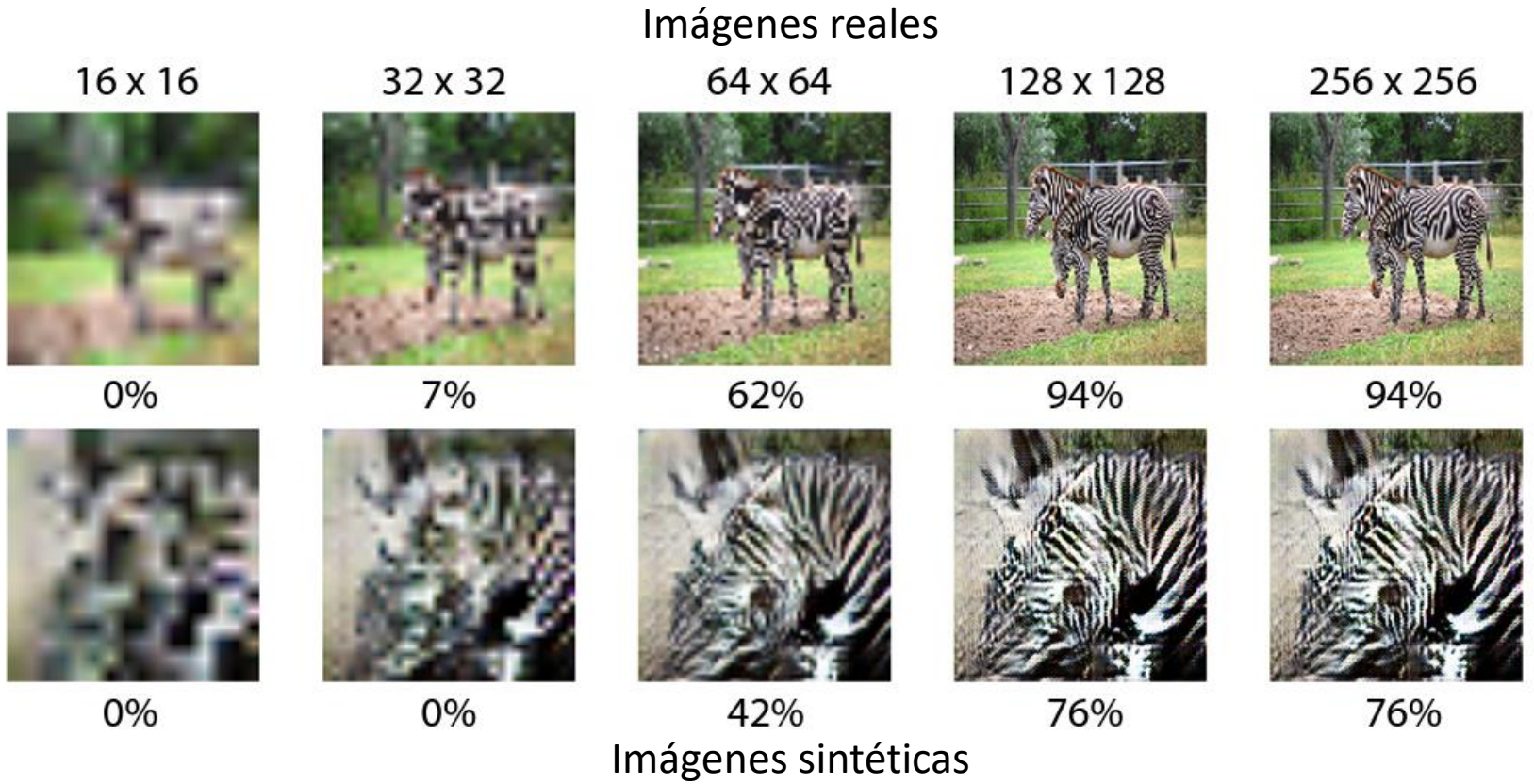
- Desventajas:
- IS usa un modelo de incepción preentrenado en ImageNet con muchas clases de objetos, por lo cual favorece modelos que generan buenos objetos (y no necesariamente datos/imágenes realistas).
- IS solo considera Pg e ignora Pr. Los resultados pueden manipularse con cambios de las imágenes reales que cambien la distribución. Como resultado podría favorecer modelos que aprenden imágenes nítidas y diversas, en lugar de Pr.
- Es una métrica asimétrica y que es afectada por la resolución de las imágenes.

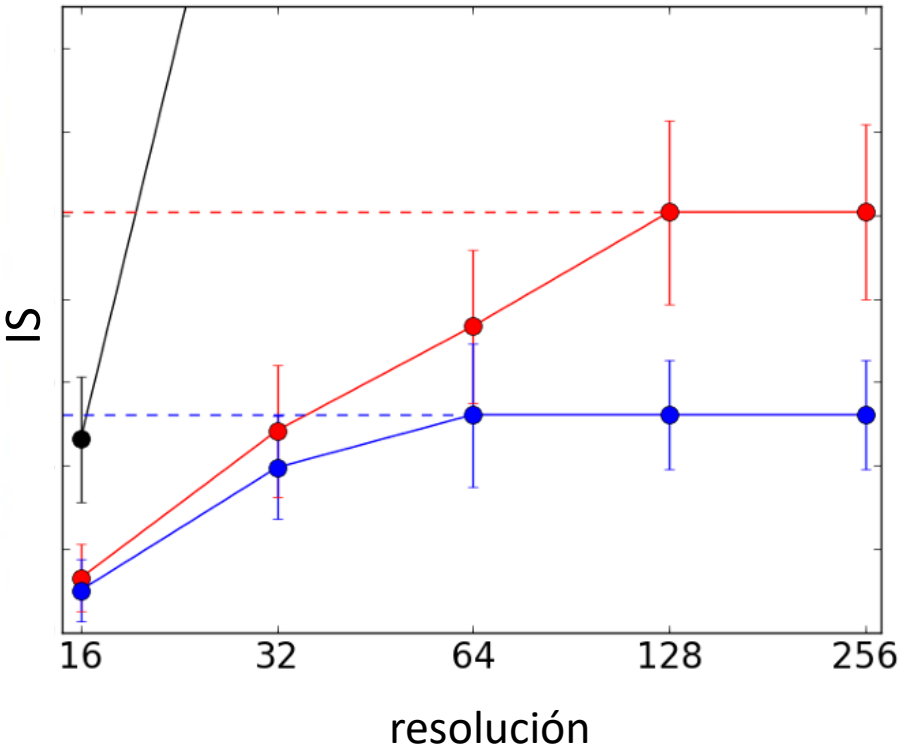
Métricas cuantitativas: inception score

- IS es agnóstico sobre el colapso de modo.
- Usualmente tiene un valor bajo de IS cuando hay colapso (es una buena propiedad de la métrica).
- Teóricamente, cuando todos los datos generados colapsan a un punto ($p(y) = p(y|\mathbf{x})$) se obtiene el mínimo valor de IS (1.0).
- Sin embargo, hay casos en que IS no mide fiablemente el colapso de modo: un modelo condicional que simplemente memorice un ejemplo por cada clase en ImageNet class tendrá un valor de IS alto.

Métricas cuantitativas: inception score

- Sensibilidad de IS respecto a la resolución de las imágenes





- Incrementar la resolución mejora la discriminabilidad (precisión)

Métricas cuantitativas: modified inception score

- Modified Inception Score (m-IS): además de asignar valores altos a modelos con baja entropía para la distribución condicional de clases sobre los datos generados $p(y|\mathbf{x})$, también evalúa la diversidad entre muestras de una misma categoría.
- Se calcula para cada clase y se reporta el valor promedio.
- m-IS evalúa la diversidad de las muestras dentro de cada clase y la calidad de las muestras.

Métricas cuantitativas: Fréchet inception distance

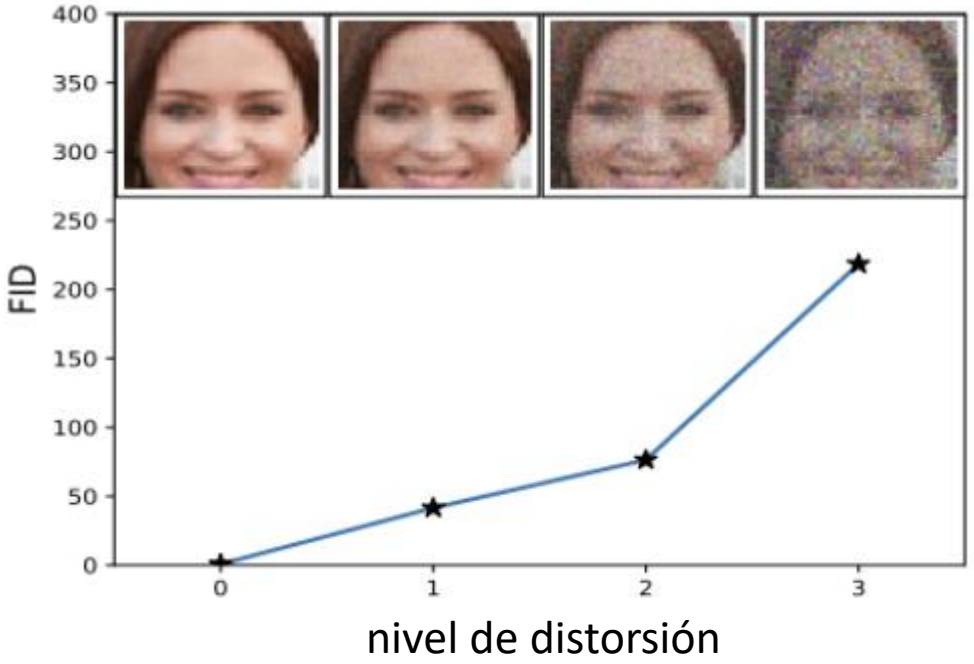
- Fréchet Inception Distance (FID) incrusta un conjunto de muestras generadas en un espacio de características dado por una capa específica de Inception Net (u otra CNN).
- Calcula “activaciones” (vectores de 2048 características) que se usan para estimar la media y la covarianza de los datos generados y de los datos reales.
- La distancia de Fréchet (Wasserstein-2) entre estos los gaussianos se utiliza para cuantificar la calidad de las muestras generadas.

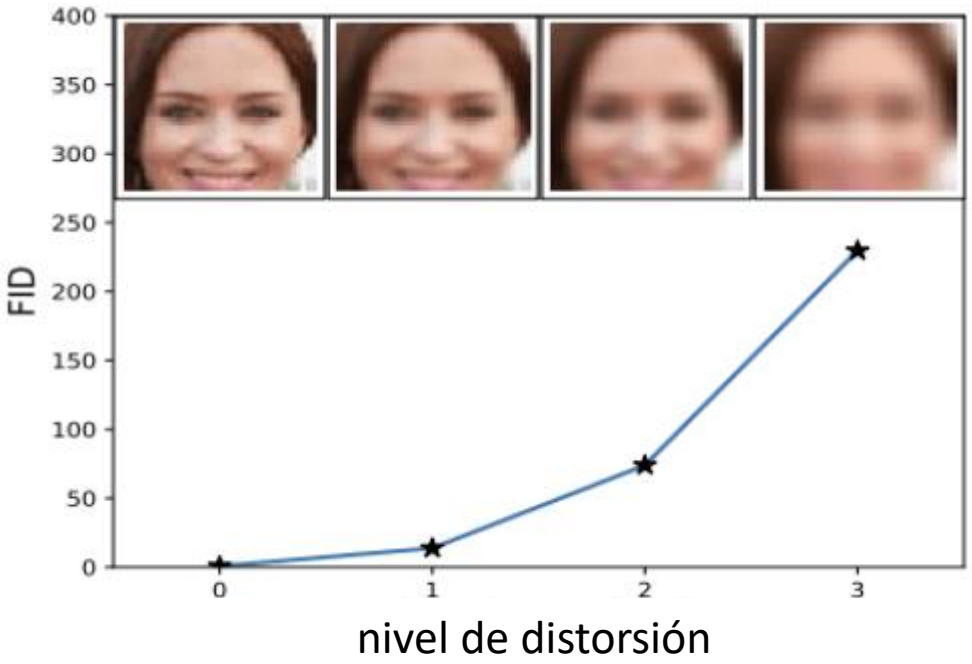
$$FID(r,g) = ||\mu_r - \mu_g||_2^2 + Tr\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}\right)$$

- Un FID más bajo significa distancias más pequeñas entre las distribuciones de datos sintéticos y reales.

Métricas cuantitativas: Fréchet inception distance

- Robusta ante distorsiones sistemáticas como ruido aleatorio o blur





Fréchet inception distance con numpy

```
def calculate_fid(act1, act2):
	# media y covarianza
	mu1, sigma1 = act1.mean(axis=0), cov(act1, rowvar=False)
	mu2, sigma2 = act2.mean(axis=0), cov(act2, rowvar=False)
	# suma de las diferencias de las medias al cuadrado
	ssdiff = numpy.sum((mu1 - mu2)**2.0)
	# raíz cuadrada del producto de covarianzas
	covmean = sqrtm(sigma1.dot(sigma2))
	# tomar la parte real para números complejos
	if iscomplexobj(covmean):
		covmean = covmean.real
	# calcular FID
	fid = ssdiff + trace(sigma1 + sigma2 - 2.0 * covmean)
	return fid
```

Fréchet inception distance en Keras

```
# preparar el modelo inception V3
model = InceptionV3(include_top=False, pooling='avg', input_shape=(299,299,3))
# definir dos conjuntos de imágenes [aleatorias]
images1 = randint(0, 255, 10*32*32*3)
images1 = images1.reshape((10,32,32,3))
images2 = randint(0, 255, 10*32*32*3)
images2 = images2.reshape((10,32,32,3))
print('Preparadas', images1.shape, images2.shape)
# convertir a punto flotante
images1 = images1.astype('float32')
images2 = images2.astype('float32')
# resize de las imágenes
images1 = scale_images(images1, (299,299,3))
images2 = scale_images(images2, (299,299,3))
print('Scaled', images1.shape, images2.shape)
```

Fréchet inception distance en Keras

```
# preprocesar images
images1 = preprocess_input(images1)
images2 = preprocess_input(images2)
# Calcular FID entre el conjunto de imágenes1 y si mismo
fid = calculate_fid(model, images1, images1)
print('FID (same): %.3f' % fid)
# Calcular FID entre el conjunto de imágenes1 y si el conjunto de imágenes2
fid = calculate_fid(model, images1, images2)
print('FID (different): %.3f' % fid)
```

- Código disponible en https://colab.research.google.com/drive/19oVAg9RHo2MwhOKW_zur1Yh0QKZuM53x?usp=sharing

Fréchet inception distance (imágenes reales)

```
from keras.datasets import cifar10
…
model = InceptionV3(include_top=False, pooling='avg', input_shape=(299,299,3))
# Cargar el dataset cifar10 (training and test datasets)
(images1, _), (images2, _) = cifar10.load_data()
shuffle(images1)
images1 = images1[:10000]
images1 = images1.astype('float32')
images2 = images2.astype('float32')
images1 = scale_images(images1, (299,299,3))
images2 = scale_images(images2, (299,299,3))
images1 = preprocess_input(images1)
images2 = preprocess_input(images2)
# Calcular FID entre ambos conjuntos
fid = calculate_fid(model, images1, images2)
print('FID: %.3f' % fid)
```

Métricas cuantitativas: Fréchet inception distance

- FID tiene buenas propiedades de discriminabilidad, robustez y eficiencia computacional.
- FID es coherente con los juicios humanos y más resistente al ruido que IS (existe una correlación negativa entre FID y la calidad visual de las muestras generadas).
- Es capaz de detectar el colapso de modo intraclase: un modelo que genera solo una imagen por clase puede tener IS alto pero tendrá un FID malo.
- A diferencia de IS, FID empeora a medida que se agregan varios tipos de artefactos a las imágenes.

Métricas cuantitativas: Fréchet inception distance

- Solo considera los dos primeros momentos de orden de las distribuciones.
- Asume que las características son de distribución gaussiana(no está asegurado).
- FID mide la distancia entre las distribuciones generadas y reales (mientras que otras métricas como IS miden la diversidad y calidad de las muestras).

Métricas cuantitativas: crítico de Wasserstein

- Aproximación de la distancia de Wasserstein
- Permite evaluar la distancia entre la distribución real y la distribución de datos generados

$$\hat{W}(\mathbf{x}_{test}, \mathbf{x}_g) = \frac{1}{N}\sum_{i=1}^{N} \hat{f}(\mathbf{x}_{test}[i]) - \frac{1}{N}\sum_{i=1}^{N}, \hat{f}(\mathbf{x}_g[i])$$

Métricas cuantitativas: crítico de Wasserstein

- Detecta sobreajuste y colapso de modo.
- Si el generador memoriza el conjunto de entrenamiento, el crítico entrenado en datos de prueba puede distinguir entre muestras y datos.
- Si se produce un colapso de modo, el crítico detectará fácilmente entre datos y muestras.
- La métrica no se saturara cuando las distribuciones no se superponen.
- La distancia de Wasserstein funciona bien cuando se calcula en un espacio de características adecuado.
- La principal limitación es su alta complejidad muestral y computacional.

Métricas cuantitativas: Classifier Two-sample Tests

- El generador se evalúa en un conjunto de test (holdout test set) que se divide en subconjuntos test-train y test-test.
- El conjunto test-train se usa para entrenar un nuevo discriminador, que intenta distinguir las imágenes generadas de las imágenes reales.
- La métrica se calcula como la precisión del nuevo discriminador en el conjunto de test-test y las imágenes recién generadas.

Métricas cuantitativas: precision y recall

- Métricas estándar en clasificación y reconocimiento de patrones, extendidas para considerar distribuciones.
- Precision mide cuánto de la distribución de los datos sintéticos puede ser generada por una parte de la distribución de datos reales.
- Recall mide cuánto de la distribución de los datos reales puede ser generada por una parte de la distribución de datos sintéticos
- Si los datos generados son similares a los reales, el valor de precision es alto (calidad de los datos generados).
- Un valor de recall alto implica que el generador es capaz de generar muestras muy cercanas a las muestras del conjunto de entrenamiento (capacidad de generación/diversidad)

Métricas cuantitativas: Performance de clasificación

- Técnica indirecta para evaluar la calidad de los algoritmos de aprendizaje no supervisados: aplicarlos como extractores de características en conjuntos de datos etiquetados y evaluar el rendimiento de los modelos lineales ajustados sobre las características aprendidas.
- Por ejemplo, para evaluar la calidad de las representaciones aprendidas por una DCGAN, se entrena el modelo en el conjunto de datos ImageNet y luego se usan las características convolucionales del discriminador de todas las capas para entrenar una SVM lineal para clasificar imágenes CIFAR-10.

Métricas cuantitativas: técnicas indirectas

- Performance de clasificación: se aplica la GAN para extraer características en conjuntos de datos etiquetados y evaluar el rendimiento de modelos lineales ajustados sobre las características aprendidas.
- Interpretabilidad semántica: usa un clasificador estándar para evaluar el realismo de los datos sintéticos. Se alimenta con datos sintéticos a una ANN entrenada con datos reales. Si el clasificador funciona bien, los datos generados son lo suficientemente precisos para discriminar la clase.
- GAN quality index: entrena un generador G con datos reales etiquetados en clases y un clasificador con datos reales. Los datos generados se alimentan al clasificador para obtener etiquetas. Un segundo clasificador CGAN, se entrena con los datos generados. GQI es el ratio de la precisión de los dos clasificadores (GQI más alto implica mejor calidad de los datos generados).

Métricas cualitativas

- Examen visual de datos generados por evaluadores humanos es una de las formas más comunes e intuitivas de evaluar las GAN.
- Ayuda a ajustar los modelos, pero tiene varios inconvenientes:
    1. Es un método costoso, engorroso y sesgado: depende de la experiencia, de la remuneración de la tarea, de la fiabilidad cuando se utiliza crowdsourcing
    2. Los resultados son difíciles de reproducir y no reflejan la capacidad de los modelos.
    3. Tiene grandes variaciones, debería utilizarse un gran número de evaluadores.
    4. Puede sesgarse fácilmente a modelos que se sobreajustan y no evaluar la diversidad o la densidad de la función de verosimilitud.
    5. En general no puede indicar si un modelo abandona un modo.

Métricas cualitativas: vecinos más cercanos

- Para detectar el sobreajuste, se agrupan muestras para visualizarlas junto a sus vecinos más cercanos en el conjunto de entrenamiento
- La métrica es muy dependiente de la distancia utilizada y tiene defectos:
  - Cuando se usa la distancia euclidiana, el método es muy sensible a perturbaciones perceptivas menores (muestras visualmente casi idénticas a una imagen de entrenamiento pueden grandes distancias euclidianas).
  - Un modelo que memoriza imágenes de entrenamiento (transformadas) puede pasar la prueba de sobreajuste de vecinos más cercanos.

Métricas cualitativas: categorización rápida de escenas

- Explotan la capacidad de los humanos de informar sobre ciertas características de las imágenes en un breve vistazo
- Prueba “de tipo Turing“, intuitiva y útil para determinar si un modelo generativo en tan buenos como la realidad.
- Problemas:
  - Alto costo
  - Condiciones experimentales difíciles de controlar en plataformas colectivas (tiempo, tamaño de la pantalla, distancia del sujeto a la pantalla, motivaciones de los sujetos, edad, estado de ánimo, retroalimentación, etc.)
  - Fallan en evaluar la diversidad de muestras generadas y pueden estar sesgadas hacia modelos que se sobreajustan a los datos de entrenamiento.

Métricas: resumen general

- Es muy difícil detectar explícitamente el sobreajuste.
- No consideran representaciones (espacios latentes) desvinculadas.
- Pocas métricas tienen cotas (inferiores/superiores).
- La concordancia entre las métricas y los juicios de percepción humana no es clara.
- Varias de las métricas más utilizadas tienen buena eficiencia muestral y computacional.

Métricas: resumen general

- Pocas métricas se enfocan en evaluar la diversidad de los datos generados.
- Algunas de las métricas más utilizadas (FID, inception score) dependen de ANN preentrenadas.
- Las métricas cualitativas en general favorecen modelos sobreentrenados y no son capaces de detectar colapso de modo.
- Se ha explorado poco la sensibilidad de las métricas a las distorsiones y variaciones de los datos de entrenamiento.