TD VI: Inteligencia Artificial

Trabajo práctico 4 (2024 2do semestre)

Encodeador de música

El objetivo de este trabajo práctico es implementar con redes neuronales un encodeador de música para obtener un vector representativo de las canciones en un espacio latente. Éstos vectores deben poder desencodearse para poder reproducir nuevamente el sonido original. Una vez obtenidos los vectores de cada canción en el espacio latente se deberán realizar análisis exploratorios para comprender y encontrar relaciones de éstos nuevos vectores.

Base de datos:

Utilizaremos la misma base de datos del TP3, GTZAN [1] que puede ser descargada desde el campus y explorado desde la pagina web de huggingface https://huggingface.co/datasets/marsyas/gtzan

La misma contiene 1000 canciones de 30sec cada una y sus clasificaciones en género: blues, classical, country, disco, hip hop, jazz, metal, pop, reggae, y rock. Para simplificar el trabajo práctico, hemos cortado los audios en 5 segundos cada uno.

Notebook ejemplo:

En éste trabajo práctico se pide crear una nueva notebook para:

1) **Encodear la canción en un vector latente (4pts)**

   En éste inciso se pide construir una red neuronal que permita encodear canciones en vectores más pequeños que luego permitan su reconstrucción. Es importante que luego de la reconstrucción pueda escuchar la canción y no perder su estructura. Realice experimentos con distintos tamaños de vector y proponga el tamaño más pequeño posible. Justifique.

2) **Análisis exploratorio de vectores latentes (4pts)**

   Al encodear las canciones en el espacio latente, esperamos poder analizar las canciones y sus relaciones en ese nuevo espacio. Realice distintos análisis que considere acorde para al menos 3 tamaños de vectores:

   - el más pequeño posible propuesto en el inciso
   - uno mas pequeño (ejemplo mitad de tamaño)
   - uno mas grande (ejemplo el doble de tamaño)

   Para este inciso puede utilizar cualquiera de los métodos no supervisados existentes, es decir no está limitado a deep learning.

   ¿Qué relaciones puede ver entre los vectores? ¿Cómo afecta la resolución?

Recuerde que para técnicas del estilo de clustering, para la selección del número de clusters se pueden utilizar técnicas como elbow, o la combinación de las métricas de homogeneidad y completitud para encontrar el balance óptimo. Otras métricas pueden encontrarse en scikit-learn:

https://scikit-learn.org/dev/modules/clustering.html#clustering-performance-evaluation

3) **Encodear música nueva (1pt)**

   Evalúen encodear música nueva, fuera de este dataset. ¿Funciona? Justificar.

4) **Generación de música (1pts)**

   Evalúe la posibilidad de realizar modificaciones sobre vectores latentes para generar nuevas canciones. ¿Es esto posible con su método?¿Por qué? ¿Qué pruebas realizó?

Ejercicios Opcionales (+1 adicional)

5) **Entrenar una red neuronal para generar canciones**

   Elija una arquitectura para generación de contenido. ¿Cuál eligió? ¿Qué resultados obtiene?

Entregables:

1- Se debe entregar un informe explicando para cada inciso los experimentos evaluados con gráficos apropiados comparativos. Por ejemplo, en el primer ejercicio, un gráfico podría mostrar la pérdida de definición a medida que disminuimos el tamaño del vector latente.

Adicional a las evaluaciones cuantitativas, se esperan evaluaciones cualitativas y sus justificaciones.

2- Notebook con parámetros definidos para: 1) poder entrenar correctamente y almacenar el modelo ganador. 2) ejecutar los análisis exploratorios, 3) evaluar una canción fuera del dataset y 4) generar la “canción” (o su intento de).

3- Canción generada. Aclaración: generar sonido “escuchable” es complejo y no será requisito para aprobar el TP.

Aclaraciones:

1- El código de la notebook incluye un `MusicDataset` que nos permite iterar sobre los elementos en el formato que necesitamos.

2- Adjuntamos también la notebook “`Cortar audios.ipynb`” donde puede encontrar código para cortar las canciones en tamaños más pequeños. Esto puede ser útil para realizar evaluaciones más pequeñas y para asegurarse que todos los audios tengan el mismo tamaño. Otro método (no provisto) puede ser bajando la resolución (down-sample).

3- Al trabajar con archivos de audios, es importante ir liberando la memoria del GPU. El ejemplo de clasificación realiza esto en cada epoch y lo puede utilizar de guía. Básicamente las líneas de código serían:

```
del wav

del genre_index

del loss

del out

torch.cuda.empty_cache()

gc.collect()
```

Entrega:

El trabajo práctico se deberá resolver de a grupos de **3 integrantes**.

Fechas y modalidad de entrega

- El **informe** y el **código** podrán entregarlos hasta el domingo 1 **de diciembre** a las 23:59:00.
- El **informe** debe ser entregado en formato **PDF**.
- Los archivos deben ponerse dentro de una carpeta llamada `tp4-gxx`, donde `xx` sea reemplazado por el número de grupo; por ejemplo, `tp4-g01`. La versión ZIP de esta carpeta debe subirse a la tarea llamada **TP4 | Entrega** en la página de la materia en el Campus Virtual.
- **Sólo 1** integrante del grupo debe realizar la **entrega**.
- Para realizar sus consultas sobre este TP, cuentan con el foro llamado **TP4 | Consultas** en la página de la materia en el Campus Virtual. Todas las dudas que surjan en relación al TP4 envíenlas exclusivamente a este foro; no usen ningún otro. Esto debe ser así porque este es el foro que está configurado de forma que los mensajes enviados lleguen únicamente al cuerpo docente y a sus compañeros de grupo. En otras palabras, si un integrante de un grupo envía una pregunta por acá, tanto esa pregunta como la respuesta, luego dada por el cuerpo docente, podrán ser vistas sólo por los integrantes del grupo en cuestión.

Bibliografía

[1] Bob L. Sturm: The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. https://arxiv.org/pdf/1306.1461.pdf