AI Lab School – Generación 3

Examen final

1- La base de datos Terravic Facial IR Database, consta de imágenes del rostro térmico de 18 individuos. Las capturas de las imágenes faciales presentan diversas variaciones, incluyendo cambios en las poses (izquierda, frontal, derecha), y el uso de distintos artefactos faciales, como lentes o gorras. En algunos casos, los rostros pueden tener tanto artefactos faciales como cambios en las poses. En la siguiente liga puedes encontrar la base de datos: https://drive.google.com/drive/folders/1HhNXDqfuXckBl8EOBi_NEJcyNIVKIcdp?usp=sharing

En función de esta base de datos y empleando tus conocimientos de machine learning aplicado a visión computacional, debes construir un modelo de reconocimiento de rostros térmicos que tiene que ser entrenado únicamente con imágenes del rostro térmico que estén libres de cualquier artefacto facial, es decir, las imágenes del conjunto de entrenamiento pueden contener variaciones en la pose, pero no pueden tener ni lentes ni gorras, tal como se muestra a continuación:

Sin embargo, los conjuntos de validación y prueba, deben tener solamente imágenes faciales térmicas con variaciones en las poses y artefactos faciales, pero no pueden tener imágenes SIN artefactos faciales. A continuación, se presentan ejemplos del tipo de imágenes que pueden contener los conjuntos de validación y prueba.

Es importante considerar que la base de datos Terravic Facial IR Database es una base de datos NO balanceada, pues el número de imágenes por clases no es el mismo. Además, de las 18 clases que constituyen la base de datos, hay 2 que únicamente tienen capturas del rostro térmico con

variaciones en las poses, pero sin artefactos faciales. Estas dos clases NO serán útiles para la construcción del modelo, así que debes descartarlas desde el principio.

Tu implementación tiene que cumplir con los siguientes requisitos:

- Características del conjunto de entrenamiento

  Este conjunto tiene que conformarse por imágenes del rostro térmico libres de cualquier artefacto facial, pero con cambios en las podes. Además, este conjunto debe ser balanceado, de tal forma que todas las 16 clases tengan el mismo número de imágenes. En particular, cada clase debe constar de 73 imágenes. Adicionalmente, es indispensable que las imágenes de cada clase tengan un orden aleatorio. Tienes que ser muy cuidadoso en este aspecto: si las imágenes de tus clases no tienen un orden aleatorio, tu respuesta no será válida y obtendrás un CERO en este problema.

- Características de los conjuntos de validación y prueba

  Como ya se describió previamente, estos dos conjuntos deben constituirse de imágenes faciales térmicas con variaciones en las poses y con artefactos faciales. Entonces, por cada clase, debes determinar cuántas de las imágenes cumplen con estas características, para después ordenarlas aleatoriamente y finalmente, asignar la mitad al conjunto de validación y la otra mitad al de prueba. Como ejemplo, si una clase tiene 201 imágenes de este tipo, debes de asignar 101 al conjunto de validación y 100 al de prueba. Al final, los conjuntos de validación y prueba serán NO balanceados, pues el número de imágenes por clase será distinto. Al igual que con el conjunto de entrenamiento, tienes que procurar que la asignación de imágenes se realice de forma aleatorio, pues si esto no es así, obtendrás un CERO en este problema.

- Obligatoriamente, debes construir los conjuntos de entrenamiento, validación y prueba mediante código, no se permite que los diseñes de forma manual, todo tiene que estar automatizado.
- Debes desplegar las gráficas de precisión y pérdida.
- Debes guardar el modelo que tenga el mejor desempeño en el conjunto de prueba, para ello, emplea ***callbacks***
- Además de utilizar la métrica ***acc*** para evaluar el desempeño de tu modelo, debes desplegar la ***matriz de confusión*** que muestre el rendimiento de tu MEJOR MODELO en el conjunto de prueba. También tienes que mostrar el valor de la métrica ***F1*** con respecto al rendimiento de tu MEJOR MODELO en el conjunto de prueba.

- Al final, tienes que compartirme en un mensaje privado de Slack, el notebook o los notebooks que usaste para resolver el problema, de tal forma que me sea posible reproducir todo, desde la construcción de los conjuntos de entrenamiento, validación y prueba, hasta la evaluación del modelo final en el conjunto de prueba y la obtención de la matriz de confusión y la métrica F1. También debes compartirme el mejor modelo que lograste.

  Los requisitos anteriores deben de cumplirse para que tengas derecho a que tu implementación reciba una calificación. Considerando lo anterior, los puntajes que puede tener tu implementación, son los siguientes:

  a) **Si la precisión en el conjunto de prueba alcanzada por tu modelo es menor a 0.90, obtendrás 0 puntos.**
  b) **Si la precisión en el conjunto de prueba alcanzada por tu modelo es mayor o igual a 0.90, pero menor a 0.95, obtendrás 30 puntos.**
  c) **Si la precisión en el conjunto de prueba alcanzada por tu modelo es mayor o igual a 0.95, obtendrás 40 puntos.**

2- El objetivo de este ejercicio, consiste en diseñar un modelo de red neuronal recurrente que sea capaz de producir texto original. Para lograrlo, primero es necesario entrenar el modelo con un conjunto amplio de secuencias de caracteres/palabras, de tal manera que cada una de estas secuencias, tenga asociada una etiqueta que represente el caracter que sigue en la secuencia. Por el ejemplo, supongamos que una secuencia de caracteres de entrada está representada por el texto “El domingo 24 de mayo de 1863, mi tío, el profesor Lidenbrock, regresó precipitadamente a su ”. Considerando dicha secuencia, su etiqueta será “c”, ya que es el caracter adecuado que la completa.

  Para este ejercicio en particular, el conjunto de secuencias de palabras que se usará, corresponde a los escritos del filósofo alemán del siglo XIX, Friedrich Nietzsche. El texto en cuestión se encuentra en el siguiente enlace: https://s3.amazonaws.com/text-datasets/nietzsche.txt

  Tu implementación tiene que cumplir con los siguientes requisitos:

  - Solamente tienes que trabajar con un conjunto de entrenamiento, no hace falta que generes más conjuntos.

- Cada instancia de entrada, ese decir, cada secuencia de caracteres, debe contener 60 caracteres.
- Cada nueva secuencia de caracteres, debe generarse 3 caracteres después del primer caracter de la secuencia anterior. Por ejemplo, consideremos el siguiente texto:

"Al principio no vi nada; mis ojos, acostumbrados a la oscuridad, se deslumbraron y se cerraron bruscamente".

Supongamos que, sólo en este ejemplo, cada secuencia se conforma por 12 caracteres, entonces, la primera secuencia sería: **[Al principio]**. Si cada nueva secuencia se genera 3 caracteres después del primer carácter de la secuencia anterior, entonces la segunda secuencia sería: **[principio no]**.

- El rendimiento de tu mejor modelo, con base a la función de pérdida, debe ser menor a 2.
- Debes guardar el modelo que tenga el mejor desempeño en el conjunto de entrenamiento, para ello, emplea ***callbacks***
- Debes desplegar la gráfica de la función de pérdida.
- Luego de entrenar tu modelo, a partir de la expresión: "philosophers, in so far as they have been very important to ", debes de agregar 200 caracteres más y mostrar la expresión final.
- Al final, tienes que compartirme en un mensaje privado de Slack, el notebook o los notebooks que usaste para resolver el problema, de tal forma que me sea posible reproducir todo el proceso. También debes compartirme el mejor modelo que lograste.

Todos los requisitos anteriores deben de cumplirse para que tu implementación pueda ser evaluada.

Valor: 30 puntos

3- El objetivo de este ejercicio, consiste en diseñar un modelo de red neuronal convolucional recurrente que sea capaz de pronosticar la temperatura del día siguiente, para lograrlo, se empleará la base de datos de la Estación Meteorológica del Instituto Max-Planck de Biogeoquímica en Jena, Alemania: http://www.bgc-jena.mpg.de/wetter/.

Esta base de datos representa una serie de tiempo, en la que 14 cantidades diferentes (incluyendo temperatura del aire, presión atmosférica, humedad y dirección del viento) fueron registradas cada 10 minutos, a lo largo de varios años. Particularmente, estaremos trabajando con una base de datos integrada por registros que abarcan el período de 2009-2016. En la terminología de las series de tiempo, cada uno de los registros de este tipo de base de datos, se

denomina ***timestep***. Además, esta base de datos, es exactamente la misma con la que trabajamos en clase, y para acceder a ella, basta que se ejecute el siguiente bloque de código:

```
!cd ~/Downloads
!mkdir jena_climate
!cd jena_climate
!wget https://s3.amazonaws.com/keras-datasets/jena_climate_2009_2016.csv.zip
!unzip jena_climate_2009_2016.csv.zip
```

Específicamente, se estará usando el archivo jena_climate_2009_2016.csv

Tu implementación tiene que cumplir con los siguientes requisitos:

- El modelo debe entrenarse con los timesteps de 10 días previos, con la consideración de que, de estos 10 días, sólo se van a tomar en cuenta para conformar el conjunto de entrenamiento, los timesteps de cada media hora, no de cada 10 minutos. La etiqueta de cada dato de entrada corresponde al timestep registrado un día después.
- El conjunto de entrenamiento debe construirse con los primeros 200,000 timesteps, mientras que los siguientes 100,000 timesteps deben conformar el conjunto de validación.
- El rendimiento de tu mejor modelo, con base a la función de pérdida del conjunto de validación, debe ser menor a 0.3
- Debes guardar el modelo que tenga el mejor desempeño en el conjunto de prueba, para ello, emplea ***callbacks***
- Debes desplegar la gráfica de la función de pérdida.
- Al final, tienes que compartirme en un mensaje privado de Slack, el notebook o los notebooks que usaste para resolver el problema, de tal forma que me sea posible reproducir todo el proceso. También debes compartirme el mejor modelo que lograste.

Todos los requisitos anteriores deben de cumplirse para que tu implementación pueda ser evaluada.

Valor: 30 puntos

Fecha límite de entrega: jueves 19 de agosto a la 1:30 am.