Predicción del Éxito de Atracciones Turísticas: Un Enfoque de Deep Learning

Práctica del Módulo de Deep Learning - KeepCoding

Miguel Angel Pardo

Introducción

Este ejercicio tiene como objetivo desarrollar un modelo avanzado de Deep Learning que permita predecir el nivel de engagement que generarán distintos puntos de interés (POIs) turísticos. Para esta tarea se dispone de dos tipos de datos: metadatos estructurados asociados a cada POI y una imagen representativa del mismo. Dentro de los metadatos hay varias variables relacionadas con el 'engagement' del POI.

Análisis exploratorio de datos (1_EDA.ipynb)

En primer lugar, se realiza una exploración general de los datos: número de muestras, número de variables, estructura de los datos y presencia de valores NA.

Variable representativa 'engagement'.

Se estudia cómo generar una variable de tipo cualitativa que represente el 'engagement' para cada muestra a partir de las variables relacionadas con el 'engagement'. Se introduce una nueva variable 'Likes_Dislikes' a partir de la diferencia entre las variables 'Likes' y 'Dislikes'. Se estudia la distribución de las variables y se observa un perfil similar en las variables 'Likes', 'Dislikes' y 'Bookmarks'. Se estudia también la correlación entre estas variables, apreciándose una fuerte correlación entre 'Likes', 'Dislikes' y 'Bookmarks'. Para evitar información redundante, se decide usar las variables 'Visits' y 'Likes_Dislikes' para calular un 'score' de 'engagement'. Se normalizan los valores de las variables 'Likes_Dislikes' y 'Visits' en el rango de 0 a 1 y se calcula el 'score' aplicando la media a estos valores. El 'score' calculado presente una distribución muy similar a unas distribución gausiana suavizada hacia valores altos de 'score'. Para asignar nivel alto o bajo de 'engagement', se elige la mediana del 'score' 0.29.

Imágenes asociadas a POI

Se realiza una representación aleatoria de imágenes de las muestras con su 'engagement score' asociado. Se plantea la posibilidad de que hubiera muestras que tuvieran imágenes que no representaran nada, por ejemplo, todo en negro o en blanco. Para buscar las muestras con este tipo de imágenes, para cada imagen se calcula la desviación estandar (SD) en cada canal RGB. Se representa la distribución de estos valores para cada canal y se observa que hay imágenes con valores bajos de SD en algún canal. Se visualizan imágenes que muestran valores de SD por debajo de 1 en algún canal y se ven claros ejemplos de imágenes 'vacias'. Se dejará al algortimo que trate con estas imágenes y ajuste el modelo teniendo en cuenta estas imágenes también.

Variable 'shortDescription'

Se analiza la variable 'shortDescription' mostrando algunos ejemplos aleatorios y dibujando la distribución del número de palabras en la variable. Para la mayoría de muestras la descripción contiene aproximadamente 19 palabras aunque hay descripciones con sólo dos palabras y hasta con 89 palabras.

Variable 'categories'

Para la variable 'categories', la mayoría de las muestras contienen tres categorías y la categoría más común es 'Historia' y la menos común es 'Gastronomía'. Al no ser muy alto el número de categorías únicas (12), esta variable se codificó como 'one hot encoding'.

Variable 'tags'

Para la variable 'tags' aparece un número muy elevado de categorías únicas (2935). Esto hace complicado el uso de esta variable y se descarta para usarla en el modelo.

Variable 'tier'

Para la variable 'tier' el valor más común es uno y el menos común es cuatro. Esta variable se codificó como 'one hot encoding'.

Variables cuantitativas

Por último, se representa la distribución de las variables contínuas 'locationLon', 'locationLat' y 'xps'

Procesado de datos (2_DLA.ipynb)

Las muestras se distribuyen en datos de entrenamiento, datos de validación y datos de test. Del total de datos, el 20% se destinan a test. El otro 80%, el 20% se dedican a validación y lo que queda a entrenamiento. La variable asociada al 'engagement' se calcula después de formar los conjuntos de datos siguiendo la metodología descrita en el Análisis Exploratorio de Datos.

Se siguen estos pasos de procesado de datos, que se aplica a los tres conjuntos de datos (entrenamiento, validación y test):

- Se genera variable con el número de etiquetas ('NumTags')
- Se codifica la variable 'categories' usando 'one hot encoder'.
- Escalado de las variables 'xps', 'locationLon' y 'locationLat'.
- Codificación de la variable 'tier' usando 'one hot encoder.

El escalado de las variables 'xps', 'locationLon', 'locationLat', 'Visits' y 'Likes_Dislikes' se hace tomando como referencia los datos de entrenamiento para evitar fenómenos de 'dataleaks'. Para la codificación de las variables 'tier' y 'categories' se utiliza como referencia el conjunto completo de datos para no perder ninguna categoría

Metodología (2_DLA.ipynb)

El esquema que se sigue para la búsqueda del modelo más útil es el siguiente:

1. Definición de parámetros y funciones: modelo con la red neuronal, optimizador, función de pérdida, parámetros fijos, parámetros a optimizar.
2. Se comprueba que el entrenamiento es correcto con las condiciones definidas usando los conjuntos de datos de entrenamiento y validación.
3. Se optimizan los parámetros mediante Optuna usando los conjuntos de datos de entrenamiento y validación.
4. Se fijan los mejores parámetros encontrados y se entrena el modelo usando el conjunto de datos de testeo para valorar la utilidad del modelo.

Para valorar la utilidad del modelo se empleó como parámetro de referencia la precisión. Sin embargo, se mostrará análisis completo de los resultados (matriz de confusión, sensibilidad, f1-score) por si fuera necesario valorar otras métricas. En este caso podría ser interesante que el modelo detectara la mayor cantidad posible de casos de alto 'engagement' (recall).

Los modelos que se tendrán en cuenta son los siguientes:

1. Fully-connected neuronal network (FCNN)
2. Convolutional neuronal network (CNN)
3. FCNN-CNN Dual-branch neuronal network (FCNN_CNN)
4. Transfer learning neuronal network (Pretrained ResNet18)
5. FCNN-ResNet18 Dual-branch neuronal network (FCNN_RN18)

En el análisis de estos modelos de partida se tratará de optimizar los hiperparámetros 'learning rate', 'batch size' y 'dropout rate'. Sin embargo, no se descarta que durante el proceso de búsqueda de un modelo mejor se pruebe la optimización de otros parámetros o la incorporación de nuevas funcionalidades.

Resultados (2_DLA.ipynb)

1. Fully-connected neuronal network.

La mejor combinación de parámetros encontrada durante la optimización da como resultado un modelo con una precisión en predicción del 66%. La optimización muestra una importancia considerable del hiperparámetro 'learning rate' sobre el resto y muy poca relevancia para el hiperparámetro 'batch size'. El modelo entrenado con los hiperparámetros optimizados se utiliza para la predicción de los valores del test. La precisión total resultante para estos datos es de 67%. La sensibilidad es alta para los POI con alto 'engagement' (0.85). Para los POIs con bajo 'engagement' la sensibilidad es baja (0.46). En la representación de la función de pérdida y el porcentaje de precisión se observa 'overfitting' ya que la predicción para el conjunto de entrenamiento llega a valores en torno al 70%.

2. Convolutional neuronal network

Para esta red la menor combinación de hiperparámetros lleva a un valor bajo de precisión mucho menor (56%) que el conseguido con la red FCNN. La mayor relevancia cae sobre el hiperparámetro 'dropout_rate' con un valor mayor al obtenido con la red FCNN. El hiperparámetro 'learning rate' presenta un valor similar al alcanzado con la red FCNN. El valor óptimo para 'batch_size', aunque no tenga tanta relevancia como 'dropout rate' es el más bajo (32). Cuando se entrena el modelo con estos hiperparámetros y se corre con el conjunto de datos de test, la precisión total resultante es del 60%, por debajo del obtenido para la red FCNN. Si se miran los valores de precisión por nivel y los valores de sensibilidad, para ambos niveles los valores son similares. En la representación de los valores de precisión según avanza el entrenamiento se hace difícil valorar la presencia de 'overfitting' debido al comportamiento inestable. Se podría tratar de buscar unos mejores hiperparámetros con una optimización permitiendo unos rangos más bajos para los hiperparámetros 'learning rate' y 'batch size'.

3. FCNN-CNN Dual-branch neuronal network.

En la optimización de los hiperparámetros para este modelo, los tres hiperparámetros optimizados tienen relevancia similar. El 'dropout rate' alcanza un valor alto muy próximo al límite marcado en la optimización. El 'learning rate' toma un valor bajo pero separado del límite inferior. El 'batch size' más óptimo toma como valor final el más alto disponible. Al final de la optimización la combinación más conveniente de parámetros hace que el modelo de una precisión para el conjunto de validación de 65%, cerca del valor obtenido con la red FCNN.

Con los hiperparámetros más convenientes se vuelve a entrenar el modelo y se aplica al conjunto de datos de test. Se obtiene una precisión total del 68%, ligeramente por encima del valor obtenido para la red FCNN. La precisión para ambos niveles de 'engagement' es similar, mientras que la sensibilidad es mayor para el nivel del 'engagement' alto.

En la representación de la función de pérdida y la precisión para los tres conjuntos de entrenamiento, validación y test se observa muy poco 'overfitting'. La precisión para el conjunto de test es muy similar a la precisión final obtenida con el conjunto de entrenamiento.

Se prueba una nueva optimización de los hiperparámetros modificando los límites de 'dropout rate' y 'batch size' para tantear la posibilidad de que sea beneficioso que tomen valores mayores. Los resultados son similares a los del experimento anterior. Se obtiene una precisión total ligeramente superior del 69%.

4. Transfer learning neuronal network (Pretrained ResNet18)

Los hiperparámetros 'learning rate' y 'batch size' en la optimización aparecen aproximadamente con la misma relevancia. Con los mejores hiperparámetros encontrados la precisión para la predicción en el conjunto de evaluación es de 64%. Con el modelo entrenado con estos parámetros se obtiene una precisión total del 62%, por debajo de otros modelos como el de FCNN y FCNN-CNN dual-branch. En la representación de la precisión se aprecia un 'overfitting' considerable llegando a alcanzar precisión valores de casi el 75% para los datos de entrenamiento

5. FCNN-ResNet18 Dual-branch neuronal network

La optimización de hiperparámetros lleva a valores que están en puntos centrales de los rangos excepto para 'batch size' que se resulta ser el valor más bajo disponible (32). Todos los hiperparámetros optimizados aparecen con igual importancia. El modelo entrenado usando los hiperparámetros más óptimos se prueba con el conjunto de test, resultando una precisión para las predicciones del 63%. La precisión para la predicción de POIs de bajo 'engagement' es mayor y la sensibilidad para POIs de alto 'engagement' es mucho mayor que para el otro tipo de POIs. Se observa un claro fenómeno de 'overfitting' con una precisión de más del 80% para el conjunto de entrenamiento y no observándose apenas mejora en la predicción del conjunto de validación a lo largo del entrenamiento.

En esta tabla se resumen los resultados obtenidos para todos los modelos y optimizaciones realizadas.

Modelo	Precisión	Precisión 0	Precisión 1	Sensib. 0	Sensib. 1	Overfitting
FCNN	0.67	0.74	0.64	0.46	0.85	Leve
CNN	0.60	0.57	0.64	0.64	0.57	No
FCNN-CNN dual-branch	0.68	0.67	0.68	0.61	0.73	No
FCNN-CNN dual-branch(2)	0.69	0.68	0.69	0.64	0.51	No
ResNet18	0.62	0.60	0.63	0.59	0.64	Importante
ResNet18-CNN dual-branch	0.63	0.69	0.61	0.40	0.84	Importante

Conclusiones

Se descarta el modelo CNN ya que es el modelo menos preciso (precisión total del 60%) y con una sensibilidad a la detección de POIs de alto 'engagement' también bajo (57%). No hay 'overfitting' por lo que se podría probar una versión más compleja de la red neuronal, aumentando capas o neuronas para intentar mejorar la capacidad de predicción del modelo.

Los modelos que incorporan una versión preentrenada de ResNet18 (ResNet18 y FCNN-ResNet18 DB) presentan un 'overfitting' muy acusado. No tienen mucha capacidad de predicción en comparación con otros modelos. Destacar la alta sensibilidad a la detección de POIs de alto 'engagement' del modelo FCNN-ResNet18 DB (84%). Para corregir este overfitting se podría probar a incorporar herramientas de regularización. Para intentar mejorar la predicción de estos modelos se podría aplicar 'fine tunning' y entrenar alguna capa de la red ResNet18.

El modelo FCNN tiene una precisión de las más altas (67%) y la mejor sensibilidad a la detección de POIs de alto 'engagement' (85%). En el resultado de su entrenamiento se obseva algo de 'overfitting'. A pesar de ser un modelo muy sencillo por su desempeño en precisión y sensibilidad podría ser una buena opción. Para intentar mejorar el modelo se podría probar a introducir algún elemento de regularización extra (reducción del 'overfitting') y se podría probar una red más compleja aumenta capas y número de neuronas (mejorar precisión).

Los mejores modelos serían los que presentan una arquitectura 'dual-branch' con una rama FCNN y otra CNN. Los dos modelos presentan los valores más altos de precisión (68% y 69%), valores altos de sensibilidad a detección de POIs de alto 'engagement' (73% y 71%) y no presentan 'overfitting'. Se podría intentar mejorar este modelo aumentando la complejidad de la redm jugando con las capas y el número de neuronas.