

Evaluación, refinamiento y comparación de modelos

Diego Arturo Padilla Domínguez
A01552594

Introducción

En este reporte se llevará a cabo la comparación de distintos modelos de “Machine Learning” que tienen un mismo fin, la clasificación de caras de 130 personajes de anime. Se utilizará como métrica de evaluación “accuracy” debido a que es la más usada en este tipo de situaciones [1].

Preprocesamiento de los datos

Se tiene un conjunto de imágenes categorizadas por el nombre de cada personaje de anime, para cada categoría hay un total de 75 imágenes, siendo una cantidad muy pequeña para un entrenamiento de un modelo y debido a que aumentar el conjunto con nuevas imágenes sería muy tardado, se llevó a cabo el aumento de los datos para el entrenamiento por medio de la modificación automatizada de los datos actuales.

Los tipos de modificación que se hicieron son los siguientes:

- horizontal_flip
- vertical_flip
- rescale (1. / 255)
- shear_range (0.1)
- zoom_range (0.1)
- width_shift_range (0.1)
- height_shift_range (0.1)
- rotation_range (30)

Se optó por estas modificaciones debido a que no hacen muchos cambios a las imágenes, esto se decidió debido a que hay muchos personajes que tienen rasgos similares y al momento de hacer cambios radicales provocaría que esa similitud fuese aún mayor y el modelo no pueda aprender a diferenciarlos de manera correcta.

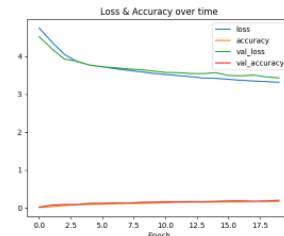
Modelo 1

En la creación de este modelo se utilizaron tan solo 5 capas básicas de keras, entre las que se encuentra:

- Conv2d (relu)
- GlobalAveragePooling2D
- Dropout
- Dense (relu)
- Dense (softmax)

El entrenamiento dio los siguientes resultados:

Precisión: 19.92%



Se puede observar que el modelo no está ‘overfited’ ya que tanto el dataset de entrenamiento como el de pruebas tienen una precisión bastante similar, siendo la diferencia cada vez mas pequeña con el paso de las épocas.

Modelo 2

En el anterior modelo se vieron los resultados con un poco cantidad de capas y esto nos dio un resultado con una precisión muy baja, por ello se creó un nuevo modelo añadiendo nuevas capas:

- BatchNormalization
- Flatten
- 2 Dense (relu)
- Dropout

El objetivo de esto es mejorar la precisión del modelo

Tras el entrenamiento, estos fueron los resultados:

Precisión: 17.1%



Se puede ver que el modelo no se encuentra ‘overfited’, debido a que la precisión de ambas partes del

dataset, entrenamiento y pruebas, son casi iguales.

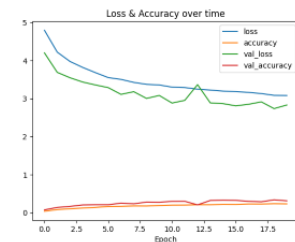
En comparación al modelo anterior a empeorado siendo una señal de que para la mejora del modelo no necesariamente sea mejor el agregar nuevas capas, esto también puede ser un indicador de que el tamaño del conjunto de datos para la complejidad del problema no sea el suficiente.

Modelo 3

Viendo el resultado de los modelos anteriores podemos concluir que solo agregando más capas al modelo este no tendrá una mejora.

Dadas las observaciones se procedió al uso de transfer learning, utilizando el modelo VGG16 debido a que es el modelo menos profundo de clasificación con mejores resultados [2], en donde este será la nueva capa de entrada. Estos fueron los resultados del entrenamiento

Precisión: 33%



Este modelo es más fluctuante entre cada una de sus épocas en comparación a los modelos anteriores, pero esto no provoca que el modelo este ‘overfited’.





El modelo ha mejorado en gran medida con respecto a los dos modelos anteriores, llegando hasta duplicar la precisión de los dos anteriores, pero sigue sin ser un modelo altamente significativo para la resolución del problema.

Nota: Todos los modelos fueron entrenados con los siguientes parámetros:

- 20 épocas
- 259 steps por época
- 44 steps de validación
- Guardada la mejor época

Comparación de modelos

Parámetro / N modelo	1	2	3
Precisión	19.9%	17%	33%
Tiempo de entrenamiento	15 minutos	20 minutos	1.5 horas

Imagen/Modelo	1	2	3
 <p>Giorno Giovanna</p>	<p>sakura_futaba - 9.27%</p> <p>giorno_giovanna - 0.83%</p> <p>(puesto 33)</p>	<p>sakura_futaba - 11.1%</p> <p>giorno_giovanna - 5.38%</p> <p>(puesto 6)</p>	<p>giorno_giovanna - 50.34%</p> <p>(puesto 1)</p>
 <p>Hatsune miku</p>	<p>hatsune_miku - 58.38%</p> <p>(puesto 1)</p>	<p>hatsune_miku - 35.25%</p> <p>(puesto 1)</p>	<p>c.c - 51.68%</p> <p>hatsune_miku - 18.86%</p> <p>(puesto 2)</p>
 <p>Asuna</p>	<p>platelet_(hataraku_saibou) - 14.9%</p> <p>asuna_(sao) - 4.73%</p> <p>(puesto 4)</p>	<p>cleveland_(azur_lane) - 9.01%</p> <p>asuna_(sao) - 2.71%</p> <p>(puesto 13)</p>	<p>asuna_(sao) - 9.94%</p> <p>(puesto 1)</p>
 <p>Aqua (konosuba)</p>	<p>xenovia_(high_school_dxd) - 40.5%</p> <p>aqua_(konosuba) - 0.91%</p> <p>(puesto 13)</p>	<p>xenovia_(high_school_dxd) - 22.16%</p> <p>aqua_(konosuba) - 2.86%</p> <p>(puesto 8)</p>	<p>platelet_(hataraku_saibou) - 6.6%</p> <p>aqua_(konosuba) - 0.84%</p> <p>(puesto 35)</p>

Conclusión

En conclusión, se compararon distintos modelos de Machine Learning para la clasificación de caras de 130 personajes de anime. Los dos primeros modelos, basados en capas básicas de Keras, no lograron obtener una precisión satisfactoria en la clasificación. Aunque se intentó mejorar el segundo modelo agregando más capas, su precisión disminuyó en comparación con el primero.

Sin embargo, el tercer modelo, que utilizó transfer learning con VGG16 como capa de entrada, mostró una mejora significativa en la precisión, alcanzando un 33%. Aunque la precisión mejoró en comparación con los modelos anteriores, aún no es lo suficientemente alta como para considerarlo una solución altamente significativa para el problema de clasificación.

Es importante destacar que, para mejorar el rendimiento de los modelos, es necesario contar con una mayor cantidad de datos en el conjunto de entrenamiento. A pesar del aumento de datos, la cantidad actual de imágenes por categoría resulta insuficiente para abordar la complejidad del problema y limita la capacidad de los modelos para aprender patrones discriminativos.

Además, se debe tener en cuenta que el entrenamiento de modelos con conjuntos de datos más grandes y complejos requerirá un mayor tiempo de entrenamiento. A medida que se aumenta el tamaño del conjunto de datos, los modelos necesitarán más iteraciones y pasos de entrenamiento para aprender y ajustarse adecuadamente a los patrones.

En resumen, se recomienda aumentar la cantidad de datos disponibles y estar preparado para invertir más tiempo en el proceso de entrenamiento con el fin de mejorar significativamente el rendimiento de los modelos de clasificación de caras de personajes de anime.

Referencias

[1] Image Classification Metrics - Getting Started with Image Classification with PyTorch. Educative: Interactive Courses for Software Developers [en línea]. [consultado el 31 de mayo de 2023]. Disponible en: <https://www.educative.io/courses/getting-started-with-image-classification-with-pytorch/xVvy4y5WAQn>

[2] Keras documentation: Keras Applications. Keras: Deep Learning for humans [en línea]. [consultado el 31 de mayo de 2023]. Disponible en: <https://keras.io/api/applications/>