**Análisis y Reporte sobre el desempeño del modelo**

Diego Emilio Barrera Hernández

A01366802

**Introducción:**

Se realizó un modelo de clasificación categórica para identificar 4 Pokémon, y con estos clasificarlos según su imagen y su nombre. Las clases son las siguientes:

* Charmander
* Squirtle
* Bulbasaur
* Pikachu

Para mejor visualización de las imágenes, se puede consultar directamente la página de referencia dónde se obtuvieron dichas imágenes, en la siguiente URL:

<https://universe.roboflow.com/olanterlab/pokemon_detector/browse>

**Método empleado**:

Para este análisis, se empleó un modelo de red neuronal basado en la arquitectura VGG16 para la clasificación de tipos de Pokémon. Se implementó un conjunto de aumento de datos durante el entrenamiento y la validación, utilizando la librería de ImageDataGenerator. La red neuronal consta de capas convolucionales, seguidas por capas densas con activación “relu” y una capa de salida con activación “softmax” para las 4 categorías de Pokémon: Squirtle, Charmander, Bulbasaur y Pikachu.

En la primera prueba del modelo, se empleó un total de 10 epocas. Durante este proceso, se observaron mejoras progresivas en la precisión tanto en los datos de entrenamiento como en los de validación. En la primera época, el modelo comenzó con un accuracy del **44.77%** en el entrenamiento y del **55.40%** en la validación. Con el transcurso de las épocas, se evidenció un crecimiento significativo en la capacidad predictiva del modelo, alcanzando una precisión del **97.08%** en el entrenamiento y del **95.40%** en la validación en la última época.

Los valores de perdida también disminuyeron de manera consistente a lo largo de las épocas, lo que indica una mejora en la capacidad del modelo para predecir con mayor precisión. Al inicio, la perdida fue de **1.2925** en el entrenamiento y de **5.9858** en la validación, mientras que al final del proceso, estos valores disminuyeron significativamente a **0.1388** en el entrenamiento y **1.0036** en la validación.

Se hicieron dos representaciones gráficas de nuestros resultados. En la Fig 1, se pueden ver los resultados obtenidos en el entrenamiento y la validación.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Fig 1: Evolución de precisión de entrenamiento y validación

En esta grafica podemos observar cómo es que la precisión de entrenamiento y de validación van evolucionando. Se puede observar que, la precisión de entrenamiento comienza en aproximadamente 0,45 y aumenta hasta aproximadamente 0,95 a medida que el modelo se entrena. La precisión de validación comienza en aproximadamente 0,6 y también aumenta hasta aproximadamente 0,95. Este es un buen resultado, pero es importante decir que la precisión de entrenamiento es ligeramente superior a la precisión de validación, esto puede ser una señal de que el modelo está sobreajustándose a los datos de entrenamiento

Ahora presentamos la Fig 2, aquí se puede observar los resultados visuales de la perdida en entrenamiento y la perdida en validación.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Fig 2: Evolución de la perdida de entrenamiento y validación

La pérdida de entrenamiento comienza en aproximadamente 0,1 y disminuye hasta aproximadamente 0,0 a medida que el modelo se entrena. La pérdida de validación comienza en aproximadamente 0,6 y también disminuye hasta aproximadamente 0,1. Como se mencionó previamente, La pérdida de entrenamiento y validación disminuye a medida que el modelo se entrena, lo que indica que el modelo está aprendiendo a realizar las tareas para las que fue diseñado.

**Mejora del modelo**:

Al igual que el primer modelo, se utilizó un modelo de red neuronal basado en la arquitectura VGG16. Solo se cambiaron algunos parámetros, como es la tasa de aprendizaje, se cambió a 0.0001 cuando anteriormente se utilizó una de 0.0002, igualmente, se cambió las características del fit del modelo, se incrementaron a 40 los steps por cada época, con ahora un total de 20 épocas; en cuanto a la validación se utilizó 25 steps.

En este caso, en la primera época, el modelo comenzó con un accuracy del **39.72%** en el entrenamiento y del **56.80%** en la validación. Con el transcurso de las épocas, se evidenció un crecimiento significativo en la capacidad predictiva del modelo, alcanzando una precisión del **98.21%** en el entrenamiento y del **96.20%** en la validación en la última época.

Los valores de perdida también disminuyeron de manera consistente a lo largo de las épocas, lo que indica una mejora en la capacidad del modelo para predecir con mayor precisión. Al inicio, la perdida fue de **1.3898** en el entrenamiento y de **5.9680** en la validación, mientras que al final del proceso, estos valores disminuyeron significativamente a **0.1589** en el entrenamiento y **1.1483** en la validación.

En la Fig 3, aquí se puede observar los resultados visuales de la predicción de este segundo modelo.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Fig 3: Evolución de precisión de entrenamiento y validación

Podemos observar, que, en este caso, la precisión es mayor al modelo anterior, en el modelo anterior se llegó a un 95% y en este mejorado a un 96%. Visualmente, se puede ver claramente como en este modelo mejorado el accuracy se estabiliza de forma más rápida que en el primer modelo, pues desde la quinta o cuarta época empieza a tener un comportamiento correcto y anteriormente desde la sexta época empezaba tener un buen comportamiento.

**Conclusión**

Comparando ambos modelos, podemos ver una mejora de **1.16%** en accuracy y una disminución de error del -0.0201, por lo que contamos con un mejor modelo, el cual, nos llega a dar mejores predicciones.

**Resultados**

<https://colab.research.google.com/drive/1GRmrDIxAFcMwTYbep9_jVtQ_LzdnakLd?usp=sharing>

|  |  |  |
| --- | --- | --- |
| **Modelo** | **Accuracy Test** | **Accuracy val** |
| **Modelo 1** | 97.08 | 95.40 |
| **Modelo 2** | 98.21 | 96.20 |