UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO INFORMÁTICA

Deep learning

2023-1

Guillermo Cabrera
Julio Godoy
Manuel Perez

Informe:

"Tarea 1"

Nicolas Esteban Parra Garcia
2019422588

Concepción, - 9 de mayo de 2023

Introducción

La clasificación de imágenes espaciales es una tarea importante en la investigación astronómica, ya que permite identificar y estudiar objetos celestes con precisión. En este trabajo, se presenta la creación de un modelo de red neuronal convolucionales para la categorización de estas imágenes. Se realizaron experimentos con arquitectura de red y técnicas de regularización para evaluar su desempeño en la clasificación de imágenes astronómicas. Además, se implementaron técnicas de preprocesamiento de imágenes para aumentar la cantidad de los datos de entrada. Los resultados muestran que la metodología propuesta permite obtener modelos de alta precisión en clasificación de las categorías especificadas.

Materiales y métodos

A continuación se presentan todos los detalles respecto a la implementación del modelo. El modelo se llevó a cabo en google collab con utilización de GPU.

Data augmentation:

- Se utilizó un **re-escalado** de imagen a 90x90 para estandarizar el tamaño de entrada.
- Se realizó el **flip** horizontal y vertical de cada imagen.
- Para cada imagen original y reflejada agrego al set de entrenamiento una copia de estas con **ruido** agregado con un factor de escala de 0.1%.

Esto efectivamente cuadruplica el tamaño original de 6431 datos de entrenamiento.

Detalles de la arquitectura

- **tamaño de batch:** 64
- **Épocas:** 60 (de las cuales se usaron 21)
- **Porcentaje del set de validación:** 10%
- **Optimizador:** Adam
- **Metricas:** RootMeanSquaredError, mean squared error, mean absolute error.
- **Función de pérdida:** MeanSquaredError
- **Método de regularización:** L1
- **Early stopping con Paciencia:** 3

Especificaciones de la red

- Este modelo contiene 9 **capas convolucionales** con 8,16,32,64 y 256 filtros respectivamente. Donde cada capa tiene un duplicado inmediatamente después exceptuando la primera. Por último la primera capa posee un filtro de 3x3, el resto de 2x2.
- Además entre las capas convolucionales, existen 4 **capas de MaxPooling** 2d de 2x2.

- Luego existe una capa de **aplanamiento** y 2 capas densas de **regularización** L1, una de 32 y otra de 64 de tamaño, ambas con un regularizador de kernel de 0.01. Luego de y para cada una, existe una capa de **dropout** del 20%.
- Por último existe una de categorización **densa** con 5 nodos, su función de activación es de softmax.
- Cabe mencionar que todas las capas convolucionales y de regularización excepto la última tienen su función de activación ReLu. El total de parámetros del modelo es de 371,413.

Resultados y discusión

A partir de los resultados entregados por el model.fit obtenemos los valores de la función de pérdida y de las 3 métricas, de estas últimas solo se interpretará el rmse.

La función de pérdida root mean squared error, inicia en 5.4338 descendiendo drásticamente en las primeras épocas, donde a partir de la sexta llegará a 0.0499 en la cual se reducirá muy gradualmente durante las épocas restantes hasta llegar a 0.0410 en la época 21. Con se puede declarar un rápido aprendizaje inicial y una estabilización rápida cuando el modelo converge, tanto el set de entrenamiento como el de validación se mantienen con tendencias muy similares.

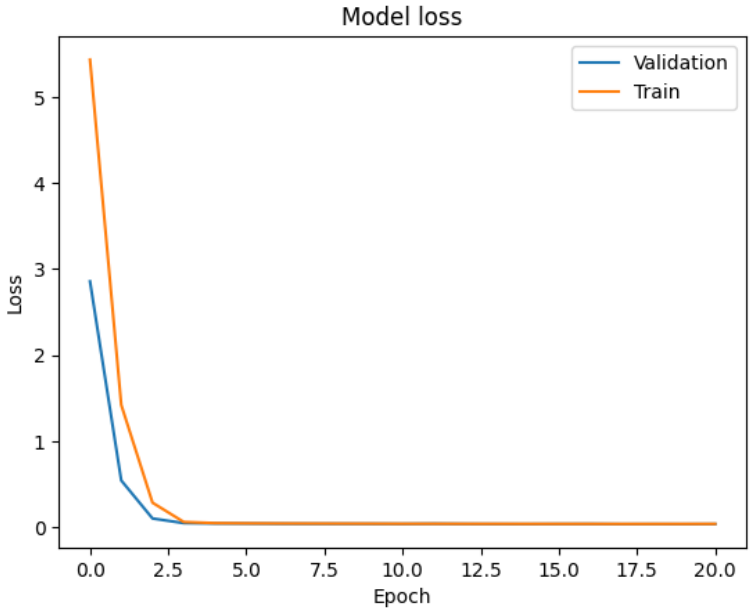

gráfico 1 "ratio de funcion de pérdida"

Por otro lado, la métrica de rmse nos indica un descenso más paulatino a lo largo de las épocas para ambos sets. La diferencia entre los valores reales y los resultados del entrenamiento logra alcanzar 0.1889 para la última época, iniciando en 0.2343

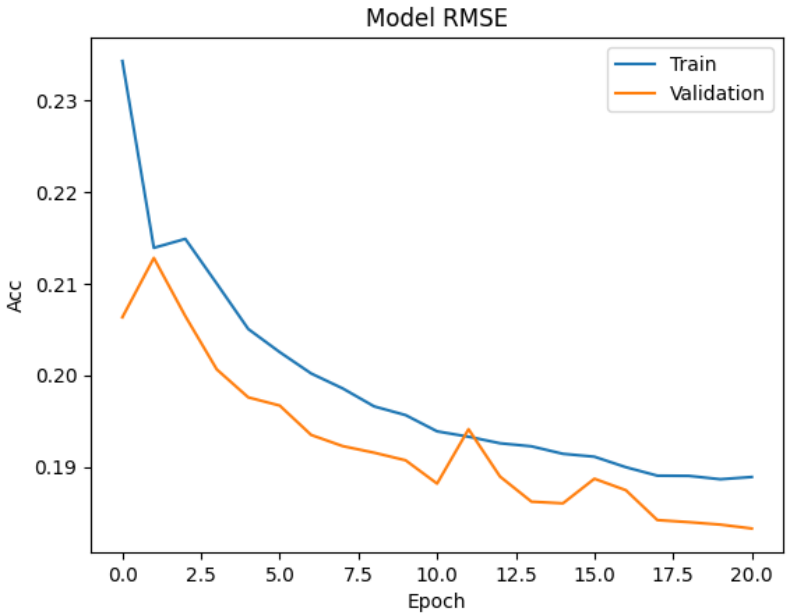

gráfico 2 "métrica de RMSE"

Con ambos gráficos se puede establecer que el entrenamiento del modelo no tiende a realizar overfitting ni underfitting, pero también, que no es óptimo debido a las grandes fluctuaciones en el set de validación y en el estancamiento final de la función de pérdida.

Observaciones

- El modelo contaba originalmente con las siguientes características adicionales que causaban un peor rendimiento, por lo que fueron removidas:
    - Data augmentation que rotaba las imágenes 90 y 270 grados, empeorando el rmse.
    - Normalización respecto al mayor valor de cada imagen, causando que el modelo no aprendiera.
    - Learning rate decay con 10000 steps y 0.96 de decay rate, causando que el modelo no aprendiera.
- En ocasiones debido al mal manejo de memoria en el data augmentation, la plataforma de google collab se quedó sin memoria RAM.

Conclusiones

Pese a no ser los resultados esperados, la efectividad del modelo no es nula, por lo que se puede concluir la formación de un modelo básico funcional. Con capacidad de realizar regresión sobre las categorías entregadas, asignando una estimación no tan lejos de la realidad.

Se espera una mejor construcción de modelo y un uso más eficiente de memoria para las proximas ocasiones.

Bibliografía

Deep Learning. (s. f.). https://www.deeplearningbook.org/ Team, K. (s. f.).
Keras documentation: Layer weight regularizers. https://keras.io/api/layers/regularizers/
Tensorflow documentation: loss functions. https://www.tensorflow.org/api_docs/python/tf/keras/losses