

Tarea 3 – Aprendizaje Automático 2024-1

Segmentación de imágenes Biomédicas U-net y U-net++

Nicolás Farfán Cheneaux

Resumen

La segmentación semántica es una tarea compleja y más aun cuando se trata de imágenes biomédicas, en el presente experimento se busca segmentar correctamente imágenes de 2 datasets combinados con múltiples imágenes de pulmones, comparando las arquitecturas U-net y U-net++ con los parámetros por defecto, obteniendo un excelente resultado muy preciso y con poco tiempo de entrenamiento al utilizar la GPU.

Introducción

Dadas las múltiples imágenes de muestra (cerca de 600) el objetivo es obtener mapas de segmentación demarcados por el color amarillo lo que fue correctamente segmentado, incorrecto rojo y faltante verde, U-net inicialmente fue una arquitectura creada para ser entrenada con pocos datos usando la aumentación de datos, por lo que al tener muchas muestras en este caso el 80% de los datos totales para entrenar es el motivo por el cual se va a tener un buen resultado.

Implementación

Se utilizo la librería **pytorch** de la mano con la dependencia **segmentation_models_pytorch** donde se encuentran la implementación de los modelos a utilizar, otro punto a considerar es que se utilizará la GPU para aprovechar su poder de paralelización y tener rapidez en el entrenamiento. Con respecto al optimizador se tomó la decisión de usar Adam dados los resultados experimentales obtenidos al entrenas distintos modelos U-net con ambos optimizadores, el optimizador SGD fue el que utilizaron los autores del paper U-net para entrenar dadas las características y limitaciones de hardware, al aplicar solamente 1 batch por imagen, sin embargo el optimizador de Adam hoy en día al poder paralelizar las tareas tiene un mejor desempeño en la pérdida obtenida tras cada época y el coeficiente de Dice (Gráfico 3).

El encoder utilizado es “resnet-34” una red residual utilizada para extraer características y fue pre entrenada con el conjunto de datos ImageNet y funciona debido a su arquitectura profunda la que permite capturar características a diferentes niveles de abstracción.

Los pesos pre entrenados del encoder “imagenet” acelera el proceso de entrenamiento y mejora la capacidad del modelo de generalizar al ajustarse al conjunto de datos a tratar, se utiliza un tamaño de batch de 32 dado que con 64 se obtiene un peor rendimiento con resultados iguales, además se tiene una sola clase al querer clasificar ausencia o presencia del elemento en la segmentación. No se utiliza ninguna función de activación dado que por sí sola la arquitectura tiene un buen resultado.

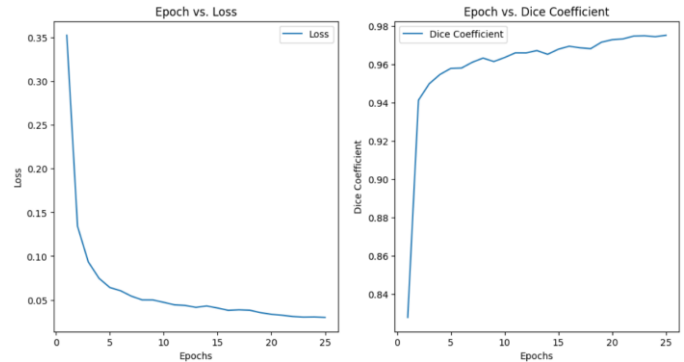
Resultados experimentales

En primer lugar, la arquitectura U-net tras entrenar por 25 épocas dada la velocidad de entrenamiento y eficiencia al usar CUDA, en la última época obteniendo una pérdida de 0.0299 y un coeficiente de Dice alto de 0.97 lo cual es una buena señal que el entrenamiento está siendo exitoso.

Epoch 1/25, Loss: 0.3527, Dice: 0.8278, Time: 5.45 seconds

Epoch 25/25, Loss: 0.0299, Dice: 0.9751, Time: 3.65 seconds

Se grafican los gráficos épocas vs pérdida y épocas vs dice.



Y se procede a realizar la predicción con los datos de prueba, se elige la tercera forma de segmentación y se toma 5 de las mejores segmentaciones obtenidas por el modelo.



Lo mismo con 5 imágenes de las peores imágenes.

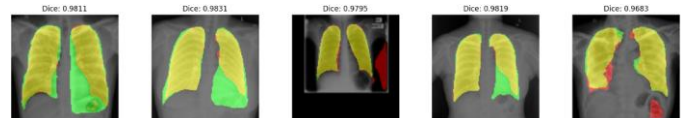
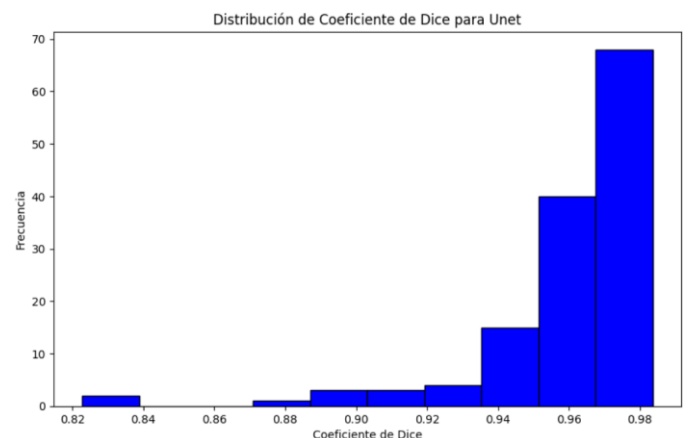


Gráfico de frecuencia y Dice U-net.



Se realiza el mismo procedimiento para el modelo U-net++ con los mismos parámetros.

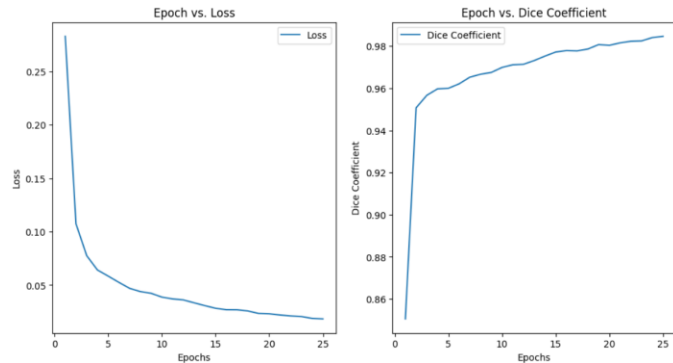
Época 1.

Epoch 1/25, Loss: 0.2828, Dice: 0.8505, Time: 7.23 seconds

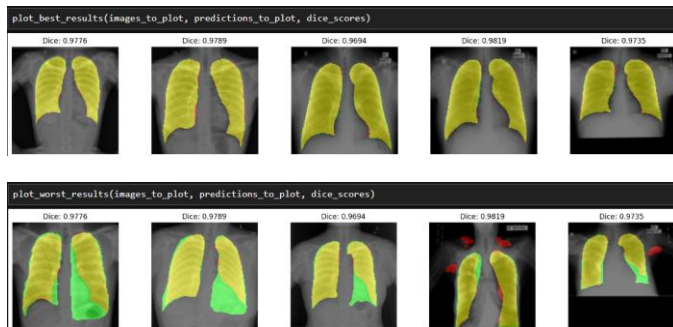
Época 25.

Epoch 25/25, Loss: 0.0184, Dice: 0.9847, Time: 7.06 seconds

Gráfico épocas vs pérdida y épocas vs dice.

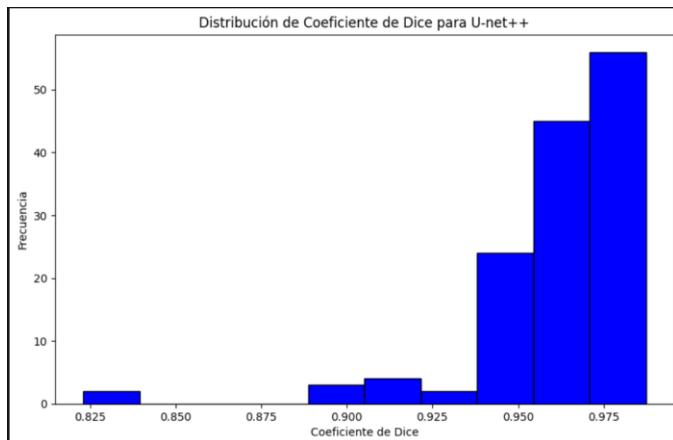


Y se procede a realizar la predicción con los datos de prueba, se elige la tercera forma de segmentación y se toma 5 de las mejores segmentaciones obtenidas por el modelo.



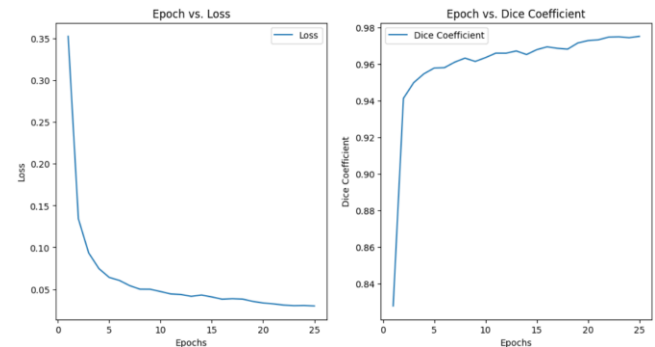
Sin embargo, se puede apreciar que de todas formas se tiene un buen resultado, si bien no hay una mejora significativa con U-net dada la naturaleza la diferencia es mínima.

Gráfico de frecuencia y Dice U-net++.

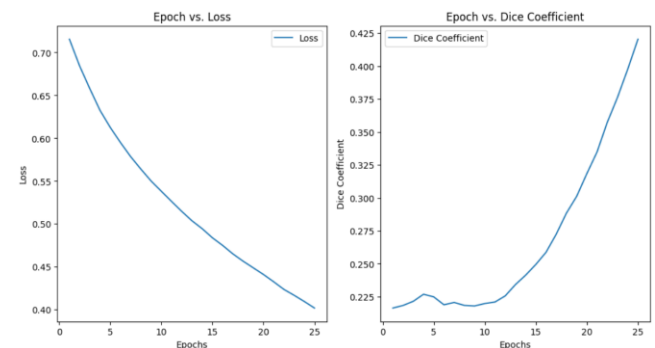


Adam VS SGD

Entrenamiento U-net usando Adam



Entrenamiento U-net usando SGD



Se puede apreciar que el coeficiente de dice alcanza máximos alrededor de 0.425 y la pérdida cae de manera exponencial pero solamente a 0.40.

Sin embargo, para Adam se obtienen resultados contrarios con una pérdida mínima alrededor de 0.03. y Dice de 0.98.

Conclusión

Al aplicar los modelos con la red residual resnet-34 y el preentrenamiento de Imagenet, lo cual es recomendado por la documentación al entrenar imágenes de esta naturaleza con 2 clases de salida, tanto U-net y U-net ++ obtienen un resultado que no se puede diferenciar en grandes rasgos uno con otro, donde si hay diferencia es al utilizar los optimizadores de **SGD** y **Adam**, por lo que dados los hiperparámetros establecidos se puede concluir que se debe escoger la arquitectura que mejor eficiencia tiene en el entrenamiento, en este caso U-net, y con respecto a las segmentaciones observadas, se puede observar más detalles incongruentes producto de la arquitectura U-net++, esto puede ser producto de la falta de ajuste de hiperparámetros pero si lo que se busca es eficiencia con un buen resultado U-net es modelo que mejor responde.