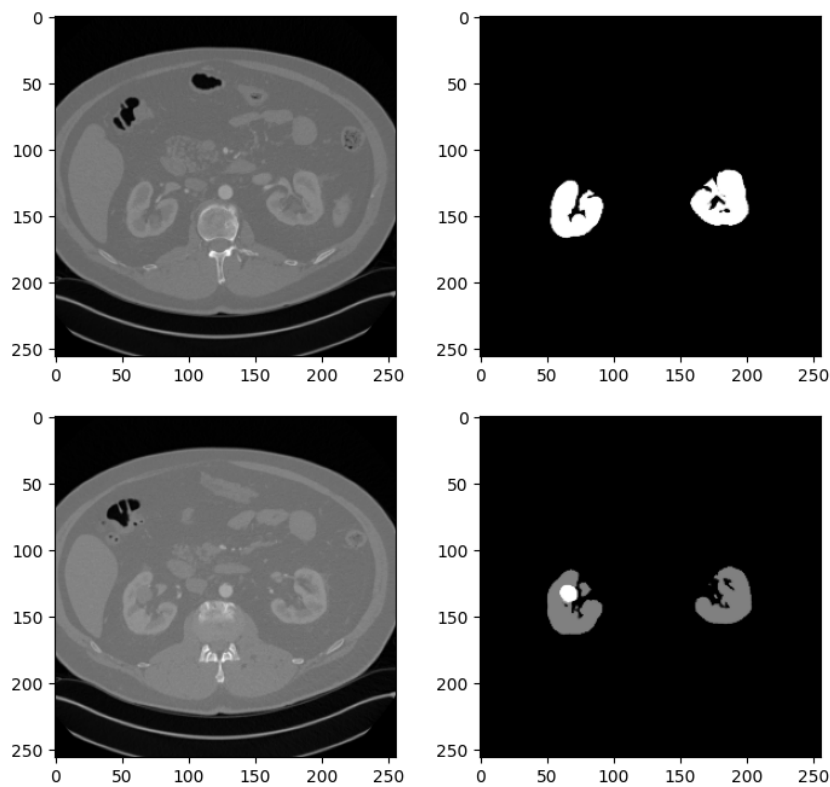


”BIOMEDICAL COMPUTER VISION COURSE”

The Kidney and Kidney Tumor Segmentation Challenge

Pucci Valentina
Politecnico di Milano, 06/06/2023



Sommario

1. Introduzione
2. Preprocessing
3. Rete Neurale
4. Risultati
5. Conclusioni

1 Introduzione

Non possedendo basi di conoscenze solide sul funzionamento delle reti neurali per la segmentazione, la rete creata è un riadattamento di quella fornita a laboratorio durante la lezione 7.

2 Preprocessing

Nella fase di preprocessing ho convertito immagini e segmentazioni da formato nifty(3D) a numpy(2D).

Il primo passo è stato ridimensionarle per riuscire ad avere immagini confrontabili tra loro, impostando una dimensione di 256x256 e 256 slices per ogni immagine 3D. Il salvataggio è stato realizzato di slice in slice identificando le 150 centrali, dopo aver verificato tramite ITK-SNAP che in media fossero quelle contenenti la maggior parte delle informazioni utili. In particolare sono state scelte le slices [50,200] per evitare di avere un numero troppo grande di immagini rappresentanti zone dove non sono presenti reni, e di conseguenza segmentazioni totalmente nere. Il programma successivamente le suddivide in set di training e set di validation (23500 e 7850 rispettivamente) numerandole in ordine crescente come (numero).npy.

3 Rete Neurale

La rete è stata realizzata sul modulo nn di Torch.

La procedura principale dell'algoritmo chiama funzioni esterne che gestiscono il loading dei dati e la realizzazione del modello, tutto in uno stesso programma.

Le procedure secondarie sono: "preprocessing()", che effettua caricamento, resizing e salvataggio delle immagini, e "My-Neural-Network()" che realizza il modello della rete neurale e si occupa di eseguire tutte le fasi di training e validation.

Il modello è stato realizzato sulla base della rete resnet101, utilizzata come pre-training, riportandola a rete per segmentazione tramite convoluzione. In merito al calcolo iterativo della loss ho scelto di utilizzare HuberLoss poichè maggiormanete si adattava al database, con Adagrad come ottimizzatore.

Le metriche di accuratezza sono calcolate con flscore.

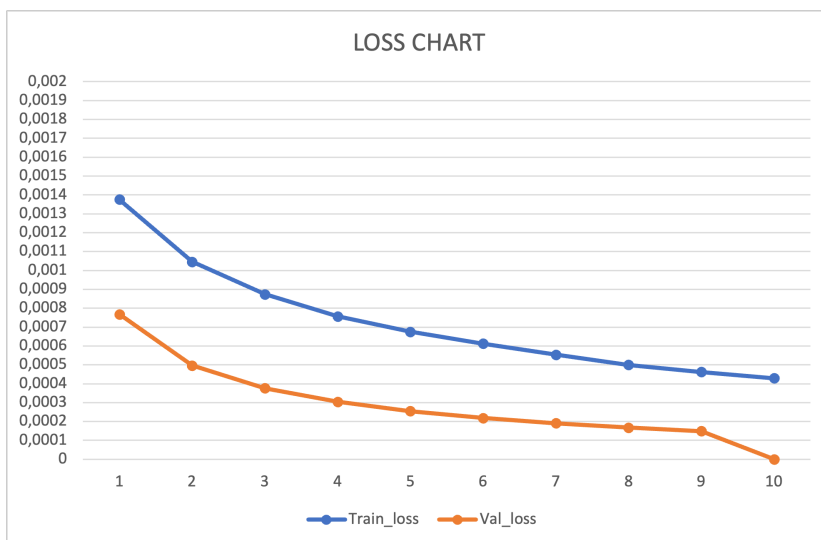
Per tenere traccia dei risultati di loss e score è stato realizzato il file log.csv [figura2]

4 Risultati

I dati sotto riportati mostrano un lungo training della rete, basato su 10 epoche con learning rate 0.001 e batch size 100 sia per immagini che segmentazioni. I valori di loss e validation sono molto bassi e hanno un andamento simile, non sembrano essere quindi presenti fenomeni di underfitting o overfitting.

epoch	Train_loss	Val_loss	Train_f1_score	Val_f1_score
1	0.001375510822981596	0.0007675132947042584	0.6581264359010669	0.6606152512998266
2	0.0010460885241627693	0.0004966126289218664	0.6633251417342744	0.6618553920656878
3	0.000874094374012202	0.00037593269371427596	0.6634043560513676	0.6620463167718246
4	0.0007575085619464517	0.00030419111135415733	0.6634730447345702	0.6619664015581886
5	0.0006754265632480383	0.00025508669205009937	0.663547474573756	0.6645896328293738
6	0.0006119916215538979	0.0002188139478676021	0.6635798959196263	0.6644304113620713
7	0.0005536982789635658	0.0001905424433061853	0.6637105412437608	0.6634432600038505
8	0.0005000432720407844	0.0001678194385021925	0.6637650818022299	0.6647620791000104
9	0.0004622085834853351	0.0001492587907705456	0.6638572629059788	0.6649212199935898
10	0.0004292736412025988	0.0001337178109679371	0.6639465293162184	0.6639569412884863

[figura2, File log.csv]



[figura3, Grafico delle loss]

5 Conclusioni

Il mio progetto non è ottimale ma sono soddisfatta delle nuove conoscenze acquisite e continuerò a lavorarci su per realizzare l'obiettivo della challenge.