Convolutional VAE

Gabriele Guidi, Nicola Bazzichi, Francesco Apruzzese

15 Gennaio 2026

1 Introduzione e descrizione del dataset

Il progetto ha lo scopo di creare un Convolutional Variational Autoencoder in grado di generare immagini di galassie appartenenti a classi differenti.
Il progetto si basa sul dataset Galaxy10 DECals, comprendente 17736 immagini di galassie suddivise in 10 classi morfologiche, tra queste sono state scelte tre classi, il più possibile diverse fra di loro al fine di distinguerle più facilmente: Round Smooth (Classe 2), Unbarred Tight Spiral (Classe 6) ed Edge-on with Bulge (Classe 9). Le immagini di partenza sono di formato 256x256, tuttavia a causa dell'alta complessità nel processare questi dati si è deciso di ridurre la risoluzione, portando le immagini a 64x64 tramite un merge dei pixel. Inoltre per evitare che il modello fosse troppo sbilanciato su un particolare gruppo, il dataset è stato bilanciato in modo che il numero di galassie per classe fosse il medesimo.

2 Architettura del modello

L'Encoder ha il compito di ridurre progressivamente la dimensionalità dell'immagine di input 64×64 pixel, attraverso una serie di N blocchi convoluzionali, caratterizzati da un *kernel* 3×3 e uno *stride* di 2. Per garantire la stabilità del training abbiamo utilizzato la *BatchNorm2d*, che normalizza le attivazioni a media zero e varianza unitaria all'interno di ciascun *batch*, per evitare che le funzioni di attivazione saturino i gradienti. Come funzione di attivazione abbiamo utilizzato la *LeakyReLU*, che permette il passaggio di piccoli gradienti anche per valori negativi, evitando il problema di neuroni spenti. Al termine di questo processo, la mappa delle caratteristiche viene appiattita e proiettata in due vettori distinti rappresentanti la media (μ) e il logaritmo della varianza ($\log \sigma^2$) della distribuzione nello spazio latente.

Per campionare un vettore dallo spazio latente si ricorre alla tecnica della riparametrizzazione. Il modello non effettua un campionamento diretto dalla distribuzione gaussiana, che renderebbe l'operazione non differenziabile, ma esprime il vettore latente z come: $z = \mu + \sigma \odot \epsilon$, dove $\epsilon \sim \mathcal{N}(0, I)$. Questo approccio introduce la componente stocastica pur mantenendo il processo differenziabile, consentendo così l'applicazione della *backpropagation* per l'aggiornamento dei pesi.

Il Decoder proietta il vettore latente in uno spazio ad alta dimensionalità e ne effettua il *reshape* per ripristinare il formato tensoriale prima dell'appiattimento. Successivamente, decomprime l'informazione tramite una serie di layer di convoluzione trasposta (*ConvTranspose2d*), raddoppiando la risoluzione spaziale a ogni passaggio fino a ripristinare la dimensione originale di 64×64 pixel. L'attivazione finale Sigmoide vincola i valori dei pixel nell'intervallo $[0, 1]$, producendo la ricostruzione finale dell'immagine o la generazione di un nuovo campione sintetico.

3 Training

Come funzione di loss si è utilizzata la KLD, pesata con un parametro beta, sommata alla SSIM che viene calcolata confrontando luminosità, contrasto e struttura di due immagini, utilizzando media, varianza e covarianza di un insieme di pixel campionati attraverso una finestra gaussiana scorrevole di dimensioni 11×11. Al fine di scegliere il modello migliore abbiamo implementato la *grid search* con questi iperparametri: batch size, learning rate, base channels, latent dimension, numero di layers e beta. Come modello finale abbiamo scelto quello che minimizzasse la somma della loss, moltiplicata per un peso, e della FID, con lo scopo di renderle confrontabili.
La FID si basa su una rete neurale già allenata (Inception-v3), alla quale passiamo le nostre immagini generate e quelle presenti nel dataset. La rete effettua quindi un confronto tra la distrubuzione delle immagini nello spazio latente misurandone la differenza. Inoltre abbiamo implementato il warm-up di beta affinchè le due loss rimanessero confrontabili durante il processo di training.

4 Metriche del Test

Nel test abbiamo utilizzato come metriche la SSIM e la FID: La prima a differenza dell'errore quadratico medio, che opera una comparazione puntuale, valuta la somiglianza strutturale tra l'immagine originale e quella ricostruita dando un risultato comparabile con l'occhio umano; la seconda invece valuta la qualità delle nostre immagini generate.

5 Risultati e considerazioni finali

Abbiamo notato come utilizzando la SSIM nella loss function al posto del MSE la rete neurale imparasse molto meglio a distinguere le galassie e le loro strutture.
Nell'equilibrio tra generazione e ricostruzione delle immagini abbiamo riscontrato due problemi principali: il primo riguarda il bilanciamento dei due termini della loss: dando più importanza alla SSIM si guadagna in ricostruzione delle immagini, ma si perde in organizzazione dello spazio latente e di conseguenza le immagini generate peggiorano. Al contrario, dando troppa importanza all'organizzazione dello spazio latente, cioè alla KLD, le immagini generate sono più verosimili, ma la rete perde la capacità di distinguere le classi. Il secondo problema è stato la dimensione dello spazio latente: nel caso in cui sia troppo grande la ricostruzione delle immagini viene migliore per via della minore compressione dei dati a discapito però della generazione, infatti a causa della *curse of dimensionality* il campionamento gaussiano standard risulta inefficace per la generazione di campioni coerenti. Al contrario uno spazio di dimensione ridotta aumenta la compressione dei dati portando al peggioramento nella ricostruzione delle immagini, migliorando però la generazione delle galassie.
Abbiamo cercato quindi il compromesso migliore tra equilibrio delle loss function e dimensione dello spazio latente, ottenendo una ricostruzione delle immagini accettabile e una generazione che rendesse distinguibili le classi di galassie.
Le galassie ricostruite (Fig. 1) mantengo le caratteristiche morfologiche della classe a cui appartengono, questo è particolarmente evidente per le classi 2 e 9 che sono le più semplici, mentre per la 6 i bracci della spirale non vengono risolti e tendono ad apparire sfocati. Inoltre anche la gamma cromatica delle immagini risulta troppo uniforme rispetto a quanto presente nel dataset. La stessa cosa può essere notata nelle immagini generate dal Decoder(Fig. 2).
Abbiamo successivamente verificato come la VAE interpolasse tra due classi di galassie (Fig. 3) e il risultato è soddisfacente.
Nei grafici della SSIM e della KLD (Fig. 4), possiamo osservare che, eccetto per la prima epoca dove β viene inizializzato a 0.0001, mantengono un valore comparabile, garantendo il compromesso tra generazione e ricostruzione.
Infine l'andamento della training loss e della validation loss è simile (Fig. 5), questo indica che il modello ha imparato senza overfitting.

Figure 1: Immagini originali vs immagini ricostruite.

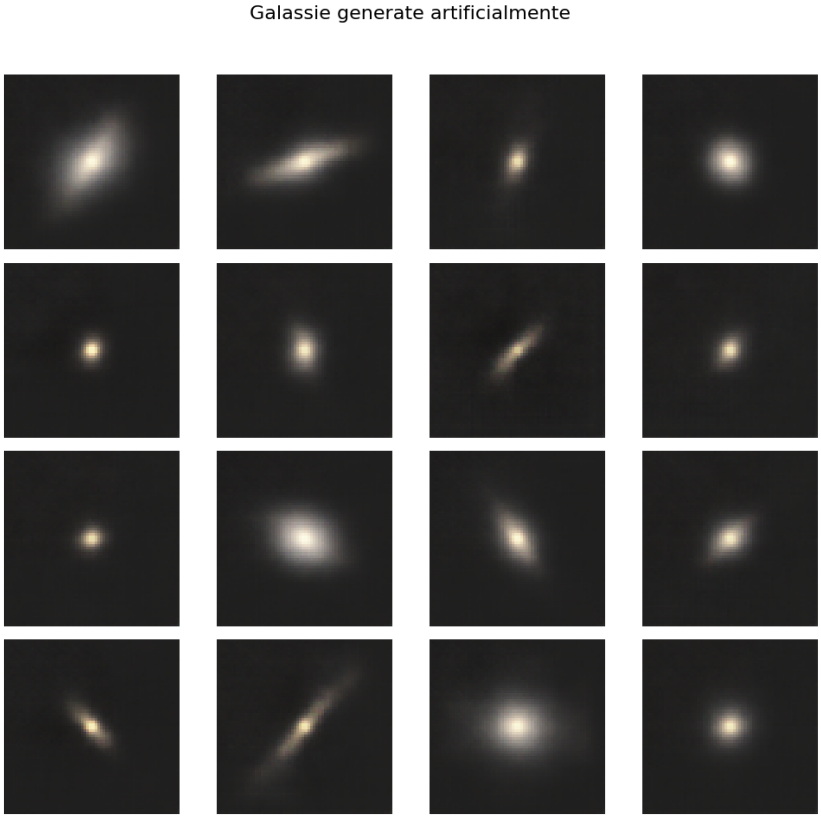

Figure 2: Immagini generate a partire da un campionamento gaussiano nello spazio latente con media nulla e varianza unitaria.

Figure 3: Metamorfosi dalla classe 2 alla classe 9, campionando tra due immagini ricostruite 8 punti nello spazio latente.

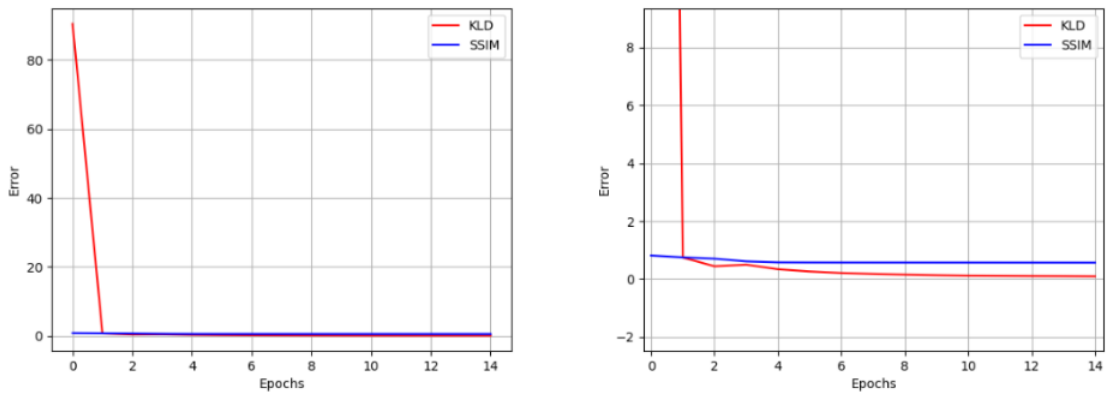

Figure 4: Grafico della SSIM e della KLD in funzione delle epoche.

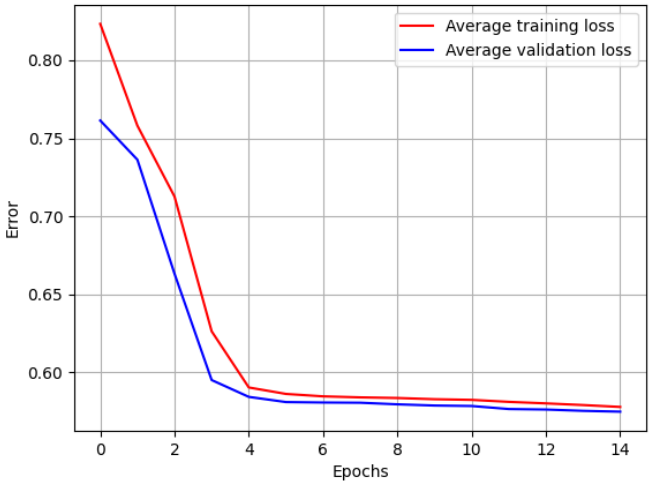

Figure 5: Average training loss e average validation loss in funzione delle epoche.