PROGETTO MACHINE LEARNING

Compressione di immagini tramite Autoencoder

Di Iorio Ludovico 0336019 – 2024/25

[ludovico.diiorio@students.uniroma2.eu](mailto:ludovico.diiorio@students.uniroma2.eu)

***Introduzione***

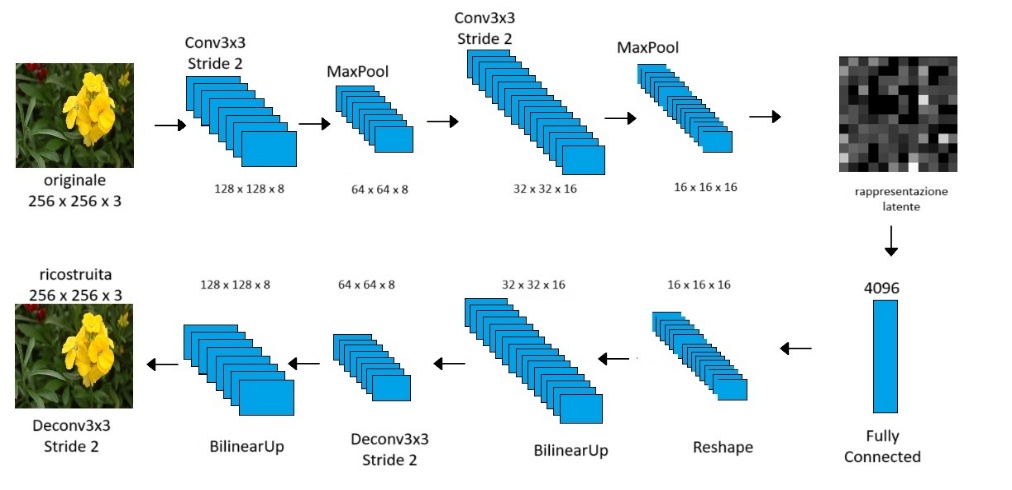
Il progetto si occupa di realizzare un autoencoder (ed eventuali varianti) che sia in grado di comprimere le immagini del dataset *102flowers* e, partendo dalla rappresentazione latente, deve essere in grado di ricostruire l’immagine originale. Le prestazioni del modello saranno confrontate in termini di rapporto di compressione e qualità della ricostruzione dell’immagine, valutata usando la metrica *Structural similarity index measure* (SSIM)*.* Inoltre, le performance del modello saranno confrontate con quelle dello standard JPEG.

Il progetto sarà composto da due Notebook distinti, ognuno dei quali si occupa di un differente aspetto del progetto.

* **Notebook 1:** viene definita l’architettura dell’autoencoder, addestrando il modello a lavorare con spazi latenti di diversa dimensione. Sono state definite, inoltre, due varianti del modello per vedere se si potessero migliorare le prestazioni.
* **Notebook 2:** il modello migliore ricavato dal Notebook 1 è stato confrontato con lo standard JPEG in termini di rapporto di compressione e SSIM.

***Notebook 1***

1. Dataset e preprocessing: il dataset è stato interamente caricato su Google Drive e grazie alla funzione *image\_dataset\_from\_directory()* è possibile ridimensionare direttamente le immagini (256x256) e suddividerle in training-set e test-set. Dopo aver ricavato il validation-set dal training-set, tutti i set sono stati normalizzati.
2. Architettura dell’autoencoder:



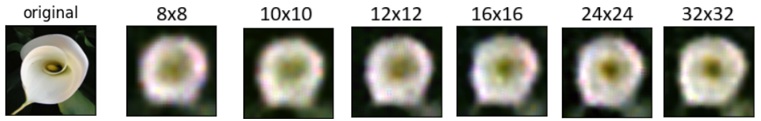
1. Addestramento: l’addestramento è avvenuto su circa l’80% dei campioni con le seguenti configurazioni:

* Batch size: 100
* Numero di epoche: 10
* Ottimizzatore: Adam
* Funzione di perdita: Reconstruction loss

1. Risultati: l’architettura descritta è stata sfruttata per generare diversi modelli, ognuno dei quali lavora con uno spazio latente di dimensione diversa.

|  |  |  |
| --- | --- | --- |
| ***Spazio latente*** | ***Loss finale*** | ***SSIM medio*** |
| 8x8 | 0.1195 | 0.34503788 |
| 10x10 | 0.1217 | 0.3397185 |
| 12x12 | 0.1176 | 0.33539808 |
| 16x16 | 0.1125 | 0.35113034 |
| 24x24 | 0.1084 | 0.3727954 |
| 32x32 | 0.1041 | 0.37024102 |

Con quest’architettura, aumentando la dimensione dello spazio latente, la ricostruzione dell’immagine migliora (anche se di poco) e ciò è dovuto al fatto che con uno spazio latente maggiore meno informazioni vengono perse nella compressione.



1. Variante 1: nel tentativo di ottenere un modello migliore, l’architettura precedente è stata modificata. Le modifiche sono state apportate per far sì che il modello sia in grado di fare una buona generalizzazione. Nello specifico, sono stati aggiunte le seguenti modifiche:

* Aggiunti due livelli di Dropout nell’encoder con probabilità di drop del 30%
* Inizializzazione dei pesi dei livelli con *He initialization*
* Aggiunti parametri di regolarizzazione L2 con parametro λ=0.01
* Data augmentation

|  |  |  |
| --- | --- | --- |
| ***Spazio latente*** | ***Loss finale*** | ***SSIM medio*** |
| 16x16 | 0.2258 | 0.29942542 |

La ricostruzione delle immagini è chiaramente peggiorata e ciò suggerisce che nel modello precedente non soffrisse di overfitting; quindi, l’introduzione di questi meccanismi ha reso l’apprendimento meno efficace, compromettendo la qualità delle immagini ricostruite.

Immagine che contiene calla, fiore, pianta, Zantedeschia

Il contenuto generato dall'IA potrebbe non essere corretto.

1. Variante 2: l’architettura dell’autoencoder è stata ridisegnata per provare ad ottenere un modello che possa ottenere ricostruzioni migliori in confronto a quelle precedenti.

Immagine che contiene testo, schermata, diagramma, design

Il contenuto generato dall'IA potrebbe non essere corretto.

Tra i livelli convoluzionali sono stati aggiunti dei livelli di normalizzazione per evitare che si vada incontro al problema del vanishing/exploding gradient, e ciò dovrebbe portare ad un miglioramento delle prestazioni. Inoltre, il numero di canali nei livelli convoluzionali è maggiore rispetto all’architettura precedente; perciò, il modello dovrebbe essere in grado di estrarre più informazioni dalle immagini, e di ricostruirne di migliori. Le configurazioni dell’addestramento sono sempre le stesse e lo spazio latente è 32x32.

|  |  |  |
| --- | --- | --- |
| ***Spazio latente*** | ***Loss finale*** | ***SSIM medio*** |
| 32x32 | 0.0938 | 0.4189291 |

***Immagine che contiene fiore, pianta, calla, petalo

Il contenuto generato dall'IA potrebbe non essere corretto.***

*N.B. alla fine del notebook 1 sono presenti ulteriori varianti nelle quali sono stati modificati altri parametri come la funzione di perdita e il numero di epoche, per vedere se si potessero ottenere risultati migliori.*

***Notebook 2***

Il modello usato per il confronto con il formato JPEG è quello della variante 2, poiché è quello che ha ottenuto risultati migliori nella ricostruzione. Le metriche usate per il confronto sono: SSIM e il rapporto di compressione, definito come il rapporto tra la dimensione in byte della codifica latente e la dimensione in byte dell’immagine originale.

L’addestramento del modello avviene solamente con immagini che hanno una determinata risoluzione (256x256), ma le immagini del dataset hanno tutte risoluzioni differenti, infatti esse sono state ridimensionate con il metodo *image\_dataset\_from\_directory()*. Ciò, ha come conseguenza che non ha senso fare un confronto tra le immagini ricostruite dal modello (256x256) e quelle originali (p.e. di dimensione 500x760), sia per il rapporto di compressione, sia per la metrica SSIM. A seguire c’è la spiegazione di come è stata gestita la situazione.

***Prestazioni JPEG***

Il confronto avviene per una singola immagine: essa è stata salvata con diversi quality factor (10, 20, …, 100) e viene confrontato il rapporto di compressione e la SSIM di queste immagini rispetto all’immagine originale.

|  |  |  |  |
| --- | --- | --- | --- |
| Quality Factor | Dimensione immagine (Byte) | SSIM | Rapporto di compressione |
| 10 | 17324 | 0.50647914 | 0.26019825773505556 |
| 20 | 26047 | 0.6277067 | 0.3912135776509462 |
| 30 | 34533 | 0.7039797 | 0.5186692700510663 |
| 40 | 37334 | 0.72767067 | 0.5607389606488435 |
| 50 | 48441 | 0.7721 | 0.7275608290778012 |
| 60 | 58359 | 0.86882514 | 0.8765244818263743 |
| 70 | 64270 | 0.9441943 | 0.9653048963652748 |
| 80 | 70453 | 0.94932264 | 1.0581706218083509 |
| 90 | 94898 | 0.9617024 | 1.4253229197957344 |
| 100 | 221373 | 0.97332007 | 3.3249173926103937 |

*\*Dimensione dell’immagine originale: 66580 Byte*

***Prestazioni modello***

Viene usata la stessa immagine utilizzata per misurare le prestazioni di JPEG. L’immagine originale (di risoluzione 500x781) è stata ridimensionata a 256x256, e ciò premette di fare il confronto tra l’immagine ricostruita dall’autoencoder e l’immagine originale.

|  |  |  |
| --- | --- | --- |
| Dimensione immagine (Byte) | SSIM | Rapporto di compressione |
| 18109 | 0.2500395 | 0.22618587442708046 |

Le prestazioni dell’autoencoder non sono comparabili con quelle di JPEG. A parità di rapporto di compressione, l’immagine generata dall’autoencoder, in termini si similarità all’originale, non è comparabile con quella ottenuta con un minimo quality factor di JPEG, il quale riesce ad comprimere l’immagine in maniera ottimale, mantenendo una buona qualità dell’immagine.

Immagine che contiene gazania, pianta, fiore, giallo

Il contenuto generato dall'IA potrebbe non essere corretto.