

Approfondimento di Intelligenza Artificiale

Studente: Tristano Munini

ANNO ACCADEMICO 2019-2020

1 Introduzione

TODO

Esempio di citazione [3].

2 GAN

Uno tra i primi metodi che permettevano di generare immagini sintetiche faceva uso di una particolare versione di *Auto-Encoder* e *Variational Auto-Encoder*. Come nel caso degli AE classici, i VAE hanno una struttura che ricorda una clessidra: la prima metà della rete permette di comprimere l'input, mappandolo in quello che viene chiamato spazio latente, di minor dimensione rispetto allo spazio di partenza; la seconda metà, invece, prende l'input compresso e lo mappa nello spazio di partenza. Durante il training si vuole ottimizzare la compressione in modo che non ci sia perdita di informazione, questo viene effettuato andando a minimizzare la distanza tra input originale ed input ricostruito. Nei VAE, in corrispondenza del punto della rete in cui si raggiunge il livello massimo di compressione (*bottleneck*), invece di essere generato il vettore compresso z , viene prodotta una coppia di vettori σ e μ che descrivono una distribuzione di probabilità dei vettori compressi. In questo modo è possibile campionare z dalla distribuzione appena prima della decompressione. Il campionamento non è un'operazione differenziabile e questo rende inapplicabile l'algoritmo della *backpropagation*, quindi risulta necessario effettuare quello che viene chiamato *reparametrization trick*. Rappresentando z come $z = \mu + \sigma \odot \varepsilon$ in cui $\varepsilon \sim \mathcal{N}(0, 1)$, quindi ε è campionata da una distribuzione normale, è possibile effettuare l'operazione di *sampling* all'esterno della rete. In questo modo μ e σ possono essere utilizzati per il calcolo del gradiente e quindi usati durante la *backpropagation*.

Dopo questa breve panoramica sulle VAE, ispirata alla lezione del MIT [2], ci si accorge che sono una soluzione astuta ma complessa e che le loro prestazioni sono vincolate strettamente allo spazio latente che si è trovato durante il training.

Le *Generative Adversarial Network* (GAN) sono state modellate appositamente per trovare un'altra soluzione al problema della creazione di immagini sintetiche. Nelle GAN sono presenti due modelli che, citando [1], “*vengono addestrati simultaneamente da un processo contraddittorio. Un generatore (“l’artista”) impara a creare immagini che sembrano reali, mentre un discriminatore (“il critico d’arte”) impara a distinguere le immagini reali dai falsi*”. Riformulando la frase si può dire che una GAN, come si vede in Figura 1, è composta da due reti: la prima viene chiamata Generatore G ed il suo scopo è fornire in output un x_{fake} che sembri appartenere alla distribuzione del *dataset* reale fornito; la seconda, detta Discriminatore D , prende in input un x_{fake} ed un x_{real} estratto dal *dataset* e deve riuscire a distinguere il dato reale da quello sintetico. Quindi il training viene svolto in quattro momenti:

- all'inizio G a partire da del rumore randomizzo genera, basandosi sulle sue conoscenze attuali, un x_{fake} ;
- successivamente D stabilisce quale tra x_{fake} ed un x_{real} fornito è sintetico;
- la prestazione di D viene valutata con il *ground truth* e con questa può essere effettuata la *backpropagation* su D ;
- con l'output del discriminatore è possibile determinare anche la prestazione di G , infatti se questo è riuscito a confondere D significa che sta raggiungendo una buona conoscenza del dominio.

Notare come G sia in grado di mappare del rumore casuale nello spazio delle *feature* del dominio e che quindi, una volta allenato, possa essere sfruttato molto facilmente: basterà avere del rumore da cui partire. Rispetto ai VAE le GAN

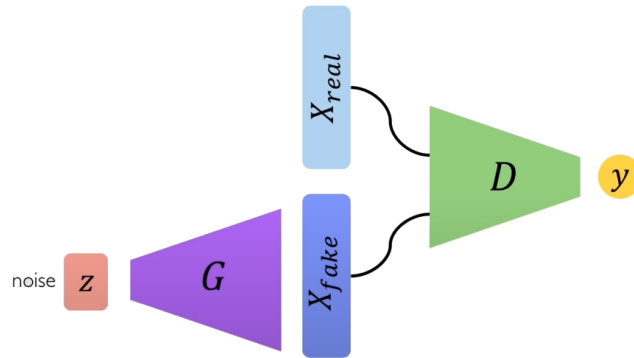


Figura 1: Architettura di una generica GAN (dalle slide in [2])

risultano non solo più intuitive ma permettono anche di raggiungere prestazioni veramente sorprendendo, come si può vedere il GAN_HD , *incui anchegliespertumanihanno*

Riferimenti bibliografici

- [1] *Deep Convolutional Generative Adversarial Network*. URL: <https://www.tensorflow.org/tutorials/generative/dcgan>.
- [2] *Deep Generative Modeling — MIT 6.S191*. URL: <https://youtu.be/rZufA635dq4>.
- [3] T. Munini. «The Monster Hunting Era». In: *The Hunter Catalog* 5 (2020), pp. 9–19.