GAN

GAN

Modelli generativi

L'ambito di lavoro dei modelli generativi è strettamente legato al concetto di **riduzione della dimensionalità**.

In termini generali, lo scopo di questi modelli è:

1. *Contribuire alla riduzione della dimensionalità dei dati.*
2. *Comprendere ed apprendere la distribuzione di probabilità del set di dati.*

Dalla distribuzione appresa, sarà potenzialmente possibile generare nuovi campioni di dati similari ai campioni di addestramento.

Tra i **modelli generativi**, sono noti:

- VAE: Variational Auto-Encoders.
- GAN: Generative Adversarial Networks.

GAN

Premessa

Le *Generative Adversarial Networks* sono reti il cui scopo è quello di generare campioni sintetici dalla distribuzione dei dati di input.

Generare perciò campioni nuovi che si avvicinino il più possibile a quelli reali.

Casi d'uso:

- *Generazione immagini sintetiche.*
- *Animazione e sintesi di video.*
- *Trasformazione di immagini.*
- *Generazione di testo e dialogo.*
- *Sintesi di suoni e musica.*
- *Sintesi di dati medici.*
- *Generazione di modelli 3D.*

GAN

Adversarial Training

Il concetto che sta alla base delle GAN è, come suggerisce il nome, l'addestramento fra due avversari, due reti neurali.

Nello specifico entrano in gioco due attori:

- **Generatore**: una rete neurale che si addestra a generare input dalla distribuzione appresa, che siano più vicini possibili agli originali.

- **Discriminatore**: una rete che si addestra a riconoscere se l'input è fake (generato) o reale eseguendo una classificazione binaria.

GAN

Adversarial Training

L'Adversarial Training si basa sul fatto che due o più reti si addestrino sfidandosi tra loro e interagendo.

I due blocchi della GAN, generatore e discriminatore «si affrontano»:

- Il ***generatore*** cerca di generare input sempre più realistici per ingannare il *discriminatore*. La generazione partirà dalla rappresentazione latente.

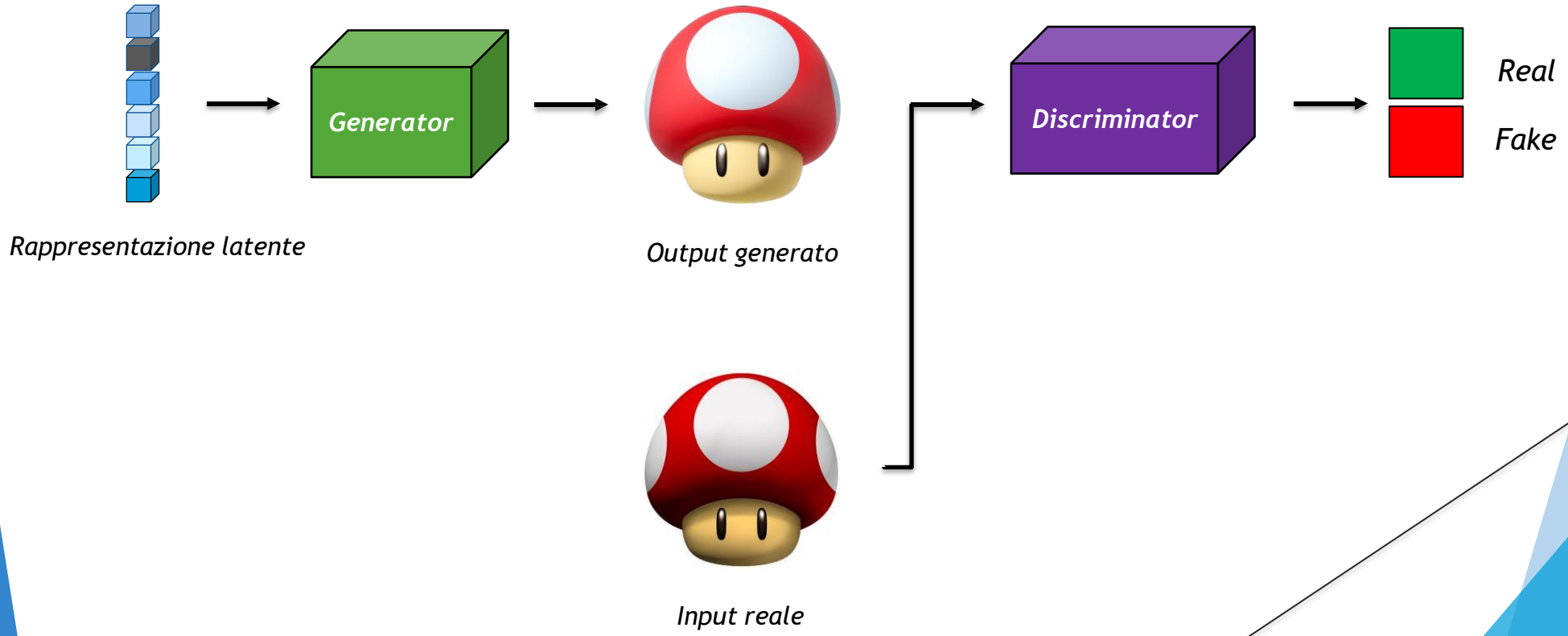
- Il ***discriminatore*** deve imparare a riconoscere quali sono gli input reali e quali quelli generati dal *generatore*.

Questo tipo di addestramento si basa sull'interazione alternata fra i due singoli modelli dei blocchi.

GAN

Rappresentazione

Generatore e discriminatore sono reti profonde. L'architettura è la seguente:



GAN

Addestramento

Il **generatore** ha raggiunto un buon risultato nel generare input fake quando il **discriminatore** non è più in grado di discriminare tra *reale* e *generato*.

Il generatore genera input *quasi perfetti*!

L'addestramento di una GAN viene eseguito in due step alternati:

1. Si addestra il **discriminatore D** mantenendo bloccato, *freezed*, il **generatore G**. Gli input fake generate da **G** sono considerate fake.

2. Si addestra il **generatore G** mantenendo bloccato, *freezed*, il **discriminatore D**. Le immagini fake generate da **G** sono considerate reali.

GAN
Addestramento

Step 1:

Si addestra **D** con un batch di dati etichettati x_{n} (reali) e x_g (fake), dove x_n sono campioni presi dal dataset di input, mentre x_g sono immagini generate da **G** con valori random della variabile latente.

Step 2:

Si addestra **G** usando l'intero modello **GAN** (***G+D***) con gli strati di **D** *freezed* (non addestrabili) con un batch di dati r_k (reali) dove r_k sono valori random della variabile latente.

GAN
Valutazione

Come si valuta un addestramento avversario?

L'obbiettivo è raggiungere l'equilibrio quando:

- **G** genera dati che per **D** siano indistinguibili dalla distribuzione dei dati di training.
- D predice sempre valori *Real* o *Fake* con probabilità di 0.5.

In poche parole:

*Il sistema funziona correttamente se entrambe le loss (sia quella di **G** che di **D**) raggiungono una convergenza e l'accuracy di **D** sta intorno al 50%.*

GAN

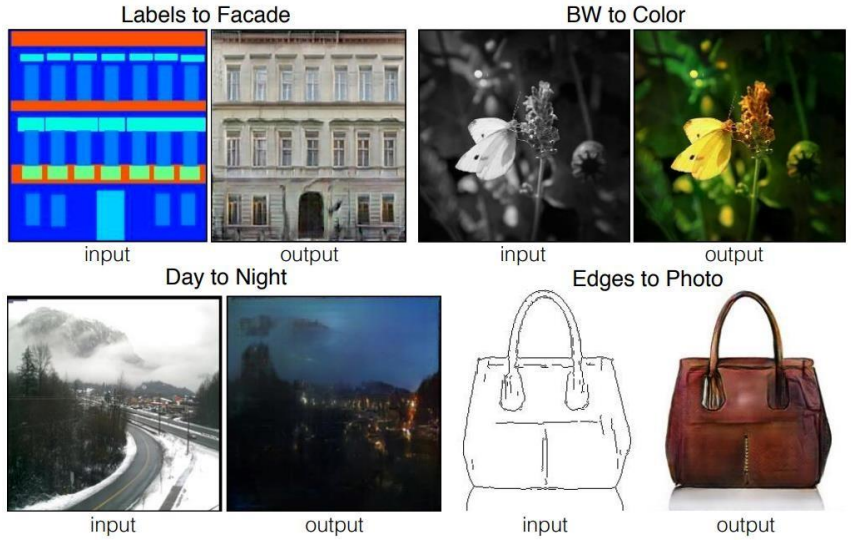
Generare nuovi dati

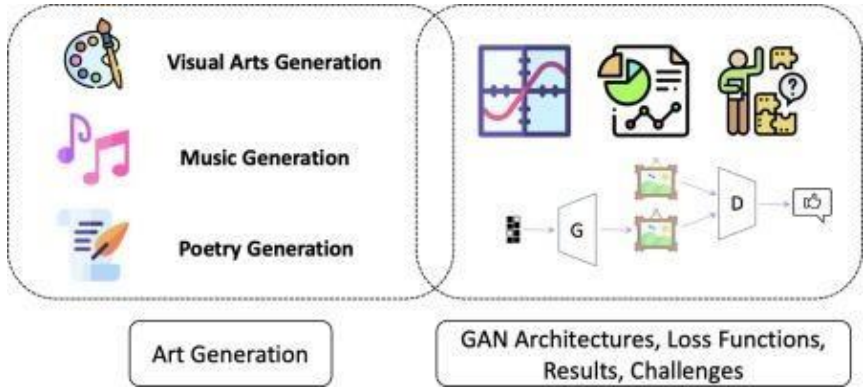
Ad addestramento terminato, si può sfruttare il modello del **generatore** per generare nuove immagini.

Il passaggio dei dati avviene quindi in una sola direzione nella rete: viene eseguito il solo *forward-pass*.

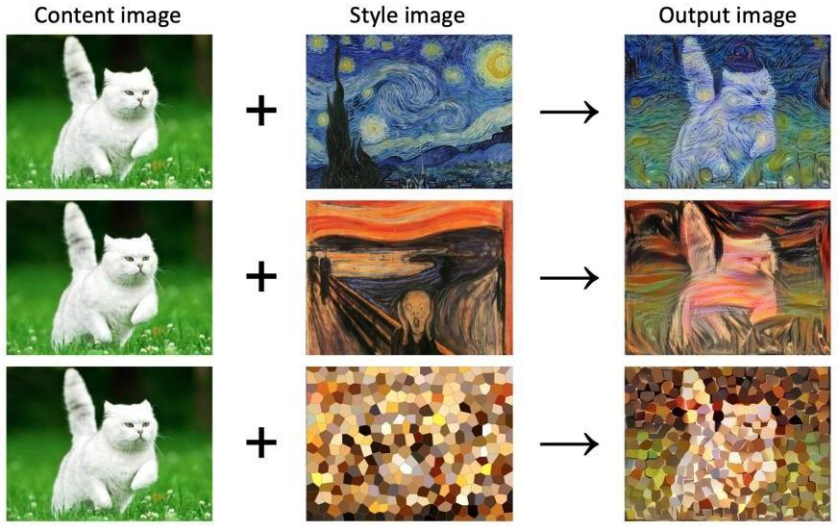
Differentemente dai **VAE**, non occorre avere un encoder che sia differenziabile per generare la rappresentazione latente z.

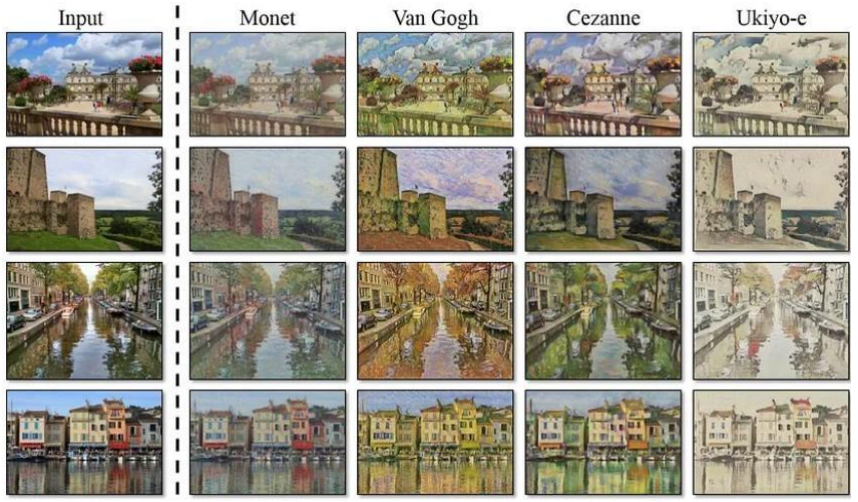
GAN
Esempi


Labels to Facade
input
output
BW to Color
input
output
Day to Night
input
output
Edges to Photo
input
output

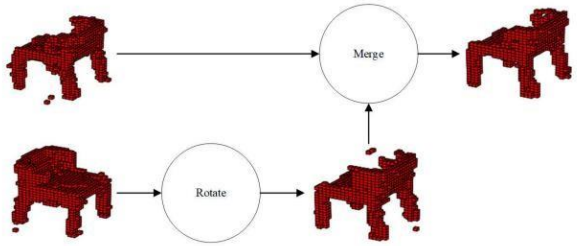

Visual Arts Generation
Music Generation
Poetry Generation
G
D
Art Generation
GAN Architectures, Loss Functions, Results, Challenges

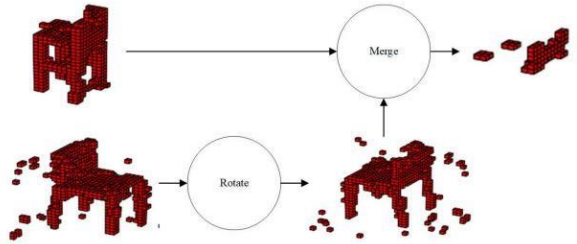
GAN
Esempi

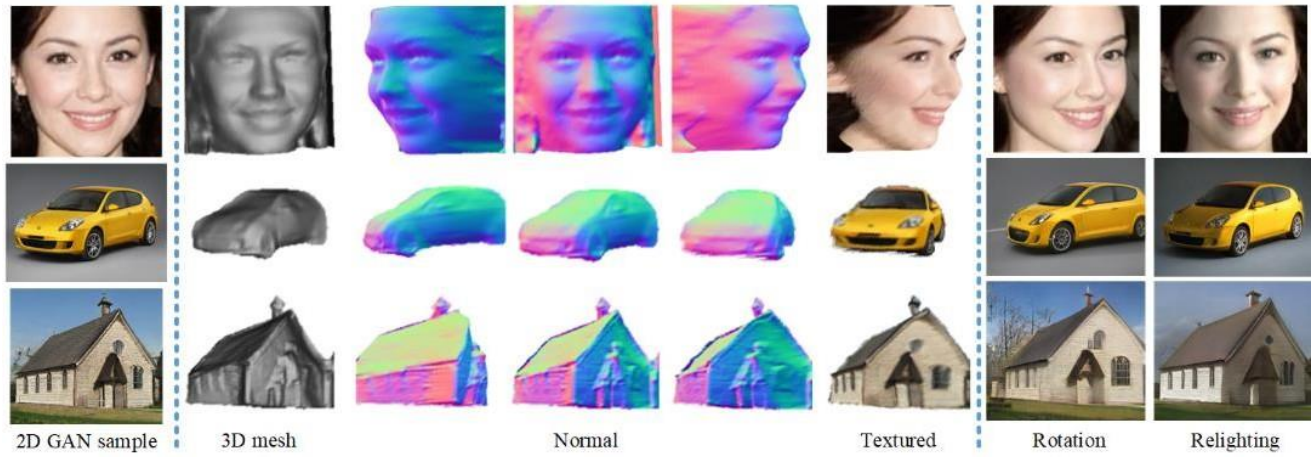

Content image
Style image
Output image


Input
Monet
Van Gogh
Cezanne
Ukiyo-e

GAN
Esempi


Merge
Rotate


Merge
Rotate


2D GAN sample
3D mesh
Normal
Textured
Rotation
Relighting

GAN

Riferimenti

Di seguito un riferimento utile per l'accesso ad esercitazioni e soluzioni sviluppate con il framework **PyTorch**:

GitHub - udacity: Projects and exercises

Di seguito un riferimento specifico ad esempi di *Gan*:

GitHub - udacity: Gan

GitHub - udacity: DCGan

GitHub - udacity: Cycle Gan

Proviamo?