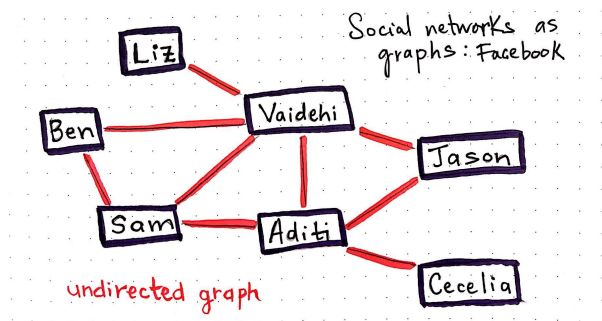
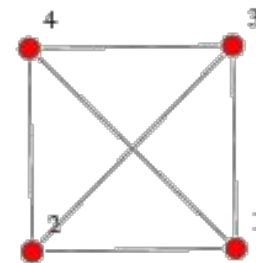
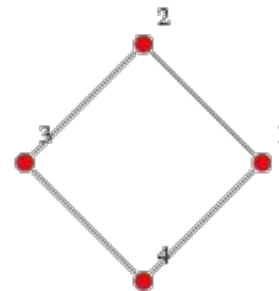
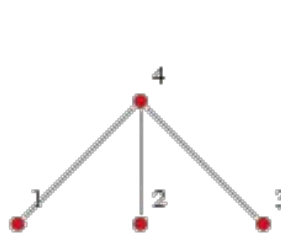


Grafi e teoria dei grafi

Un grafo è definito come un **set di vertici V** ed un **set di coppie di vertici E** (ordinate e non). Una coppia non ordinata viene chiamata **edge**, una coppia ordinata **arc**.

Per definire un grafo si può utilizzare una **matrice di adiacenza A**.

Il grado di un vertice è dato dal suo numero di edge incidenti.



$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

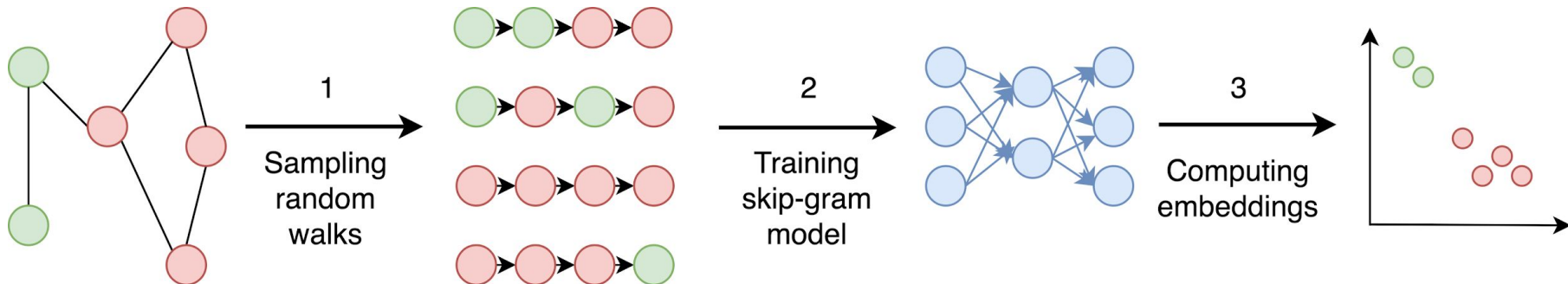
$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Graph embeddings

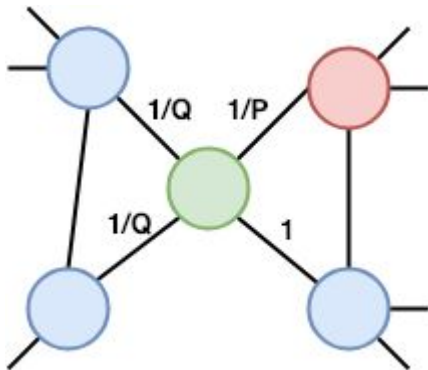
Gli embedding sono le **trasformazioni delle proprietà dei grafi ad un vettore o ad un set di vettori**. Devono cercare nel migliore dei modi di sintetizzare le informazioni della topologia del grafo, delle relazioni tra vertici ed altre informazioni.

Esistono diversi metodi, ad esempio **DeepWalk**: attraverso un algoritmo di random walk che parte da un vertice ed esplora nelle vicinanze i collegamenti con gli altri vertici in modo randomico

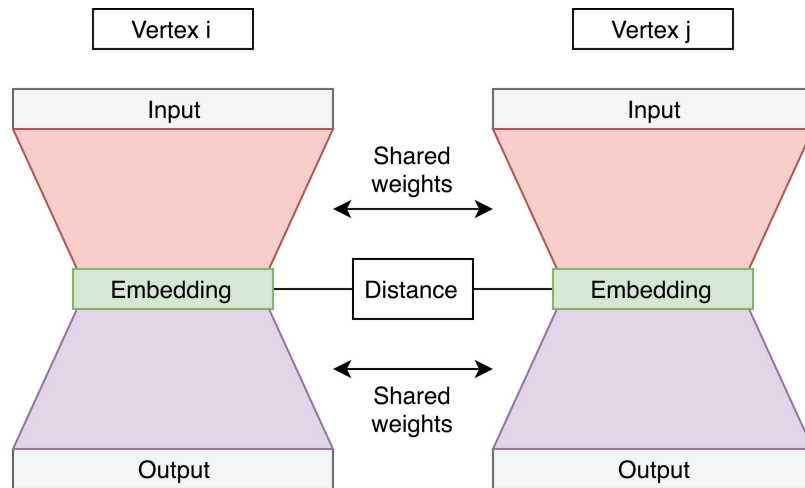


Graph embeddings

Altri metodi basati sempre su rappresentazioni vettoriali “standard” sono **node2vec** e **SDNE** (Structural Deep Network Embeddings), che cercano di sintetizzare le informazioni sulla struttura locale e globale del grafo.



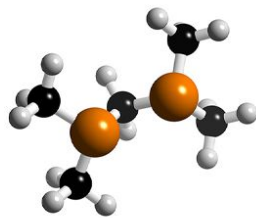
node2vec



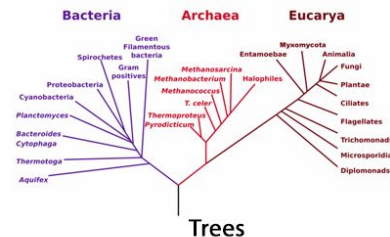
SDNE

Graph embeddings in spazi non euclidei

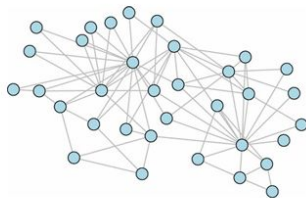
La maggior parte del deep learning si concentra su dati euclidei (i.e. immagini, testo, audio, etc...) e il fulcro di tutte le operazioni è il **vettore**, o genericamente il **tensore**. I dati non euclidei possono rappresentare oggetti e concetti più complessi con più precisione rispetto ad una rappresentazione 1D o 2D (ad esempio, grafi e varietà).



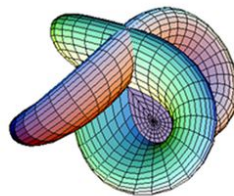
Molecules



Trees



Networks

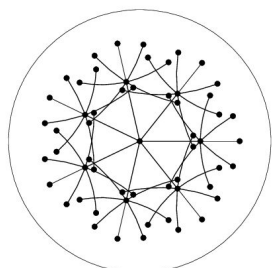


Manifolds

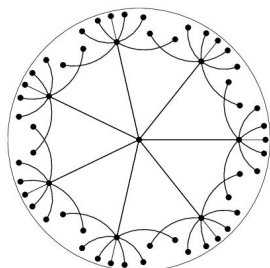
Graph embeddings in spazi non euclidei

L'obiettivo dell'embedding è di preservare distanze e altre relazioni più complesse.

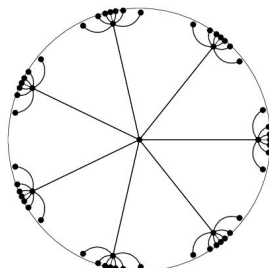
Uno spazio iperbolico riesce a definire un embedding migliore di uno spazio euclideo per dati strutturati in modo gerarchico (ad esempio alberi).



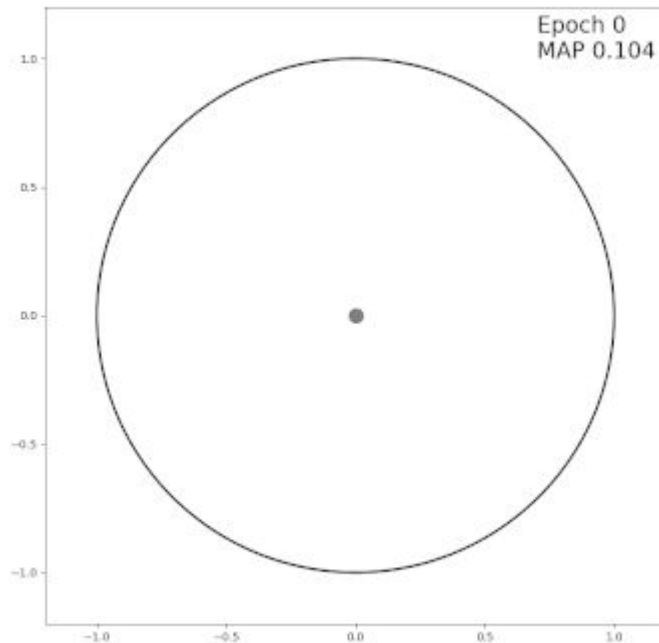
Radius 1



Radius 2



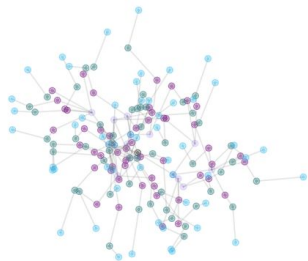
Radius 3



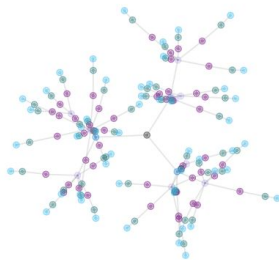
Hyperbolic Graph Convolutional Neural Networks

L'utilizzo di embedding iperbolici porta a netti miglioramenti anche nel caso delle graph convolutional networks (GCN), dove hanno permesso di ottenere lo stato dell'arte.

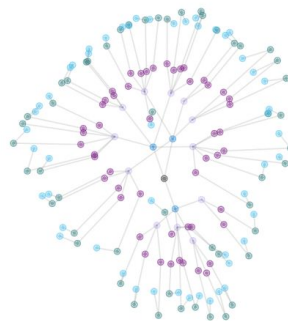
GCN First Layer



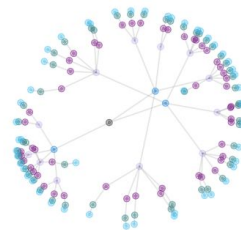
GCN Last Layer



HGCN-P First Layer

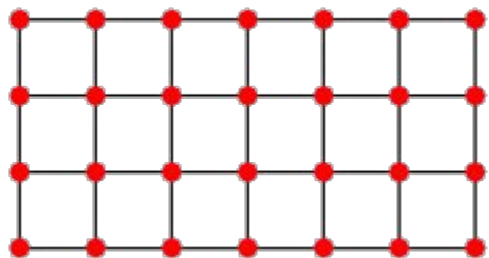


HGCN-P Last Layer



CNN e GNN

Il punto di forza delle CNN e delle RNN è la loro capacità di saper sfruttare al meglio la conoscenza delle interconnessioni fra i dati in input. Ad esempio un filtro convolutivo si basa sul fatto che i dati necessari ad elaborare un singolo pixel (per estrarne una feature) si trovino nei pixel a lui vicini (tipicamente, a due o tre pixel di distanza) e che i pixel più distanti possano essere ignorati. Da qui si può evidenziare come dietro questo concetto vi sia un grafo, in quanto la vicinanza dei pixel equivale a rappresentare l'immagine come un grafo dalla struttura perfettamente regolare:



Sfruttare questa informazione di vicinanza tra i pixel è il cuore di una rete convolutiva, sebbene i pixel in un'immagine sono interconnessi in modo estremamente regolare. C'è modo per sfruttare l'informazione contenuta nel grafo senza senza sacrificare efficienza o flessibilità delle architetture nel caso di grafi irregolari (i.e. con nodi quasi isolati, alcuni centrali etc...)?

Graph Convolutional Network

$$g(X) = \text{ReLU}(XW)$$

CNN regolare



$$g(X) = \text{ReLU}(AXW)$$

GCN, con aggiunta A matrice di adiacenza