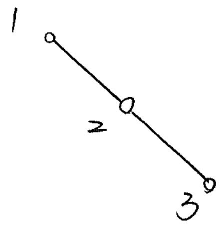
Graph theory : matrices

An easy way to store a graph in computer is by matrices.



adjacency matrix

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

Laplacian matrix

$$\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

Applications:

connecting walks

basic structure

lattice

counting spanning trees

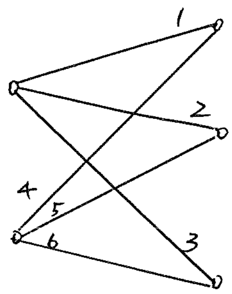
graph partition

Laplacian eigenmap

spectral clustering

Use König graph to represent a matrix.

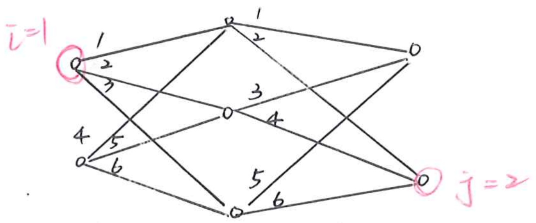
$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$



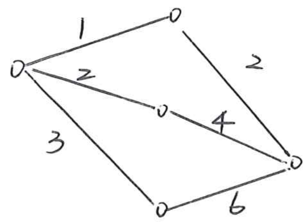
Use König graphs to do matrix multiplication.

- weight of a path is $a \cdot b \cdot c$.

- The ij-entry of AB is the sum of weights of all paths from i to j.



$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} = \begin{pmatrix} 22 & 28 \\ 49 & 64 \end{pmatrix}$$



$$\Rightarrow 1,2\text{-entry} = 1 \cdot 2 + 2 \cdot 4 + 3 \cdot 6$$
$$= 2 + 8 + 18$$
$$= 28$$

A neural network is (more or less) applying König graphs several times.