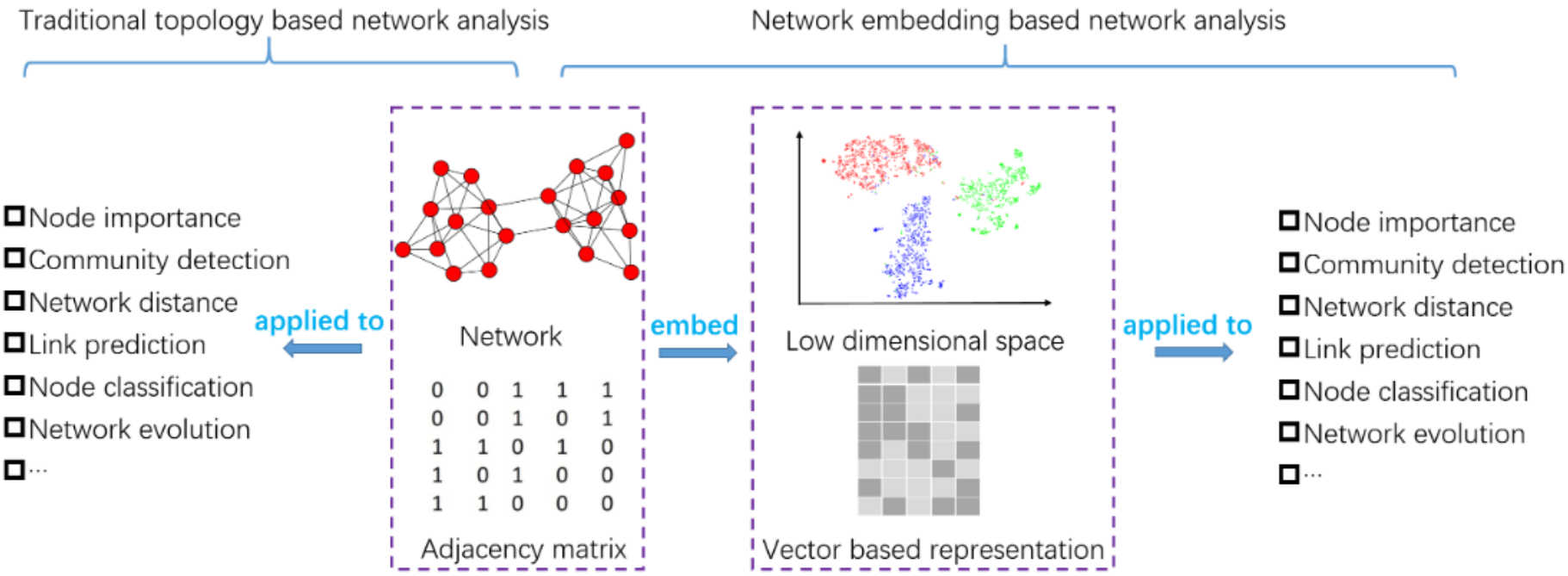
Complex Networks miscellaneous

2019.1.31(Thu)

contents

- Network embedding
- Deep learning & networks
  - Compression of deep neural networks
  - Graph neural networks
  - Community detection with autoencoders
- Some references

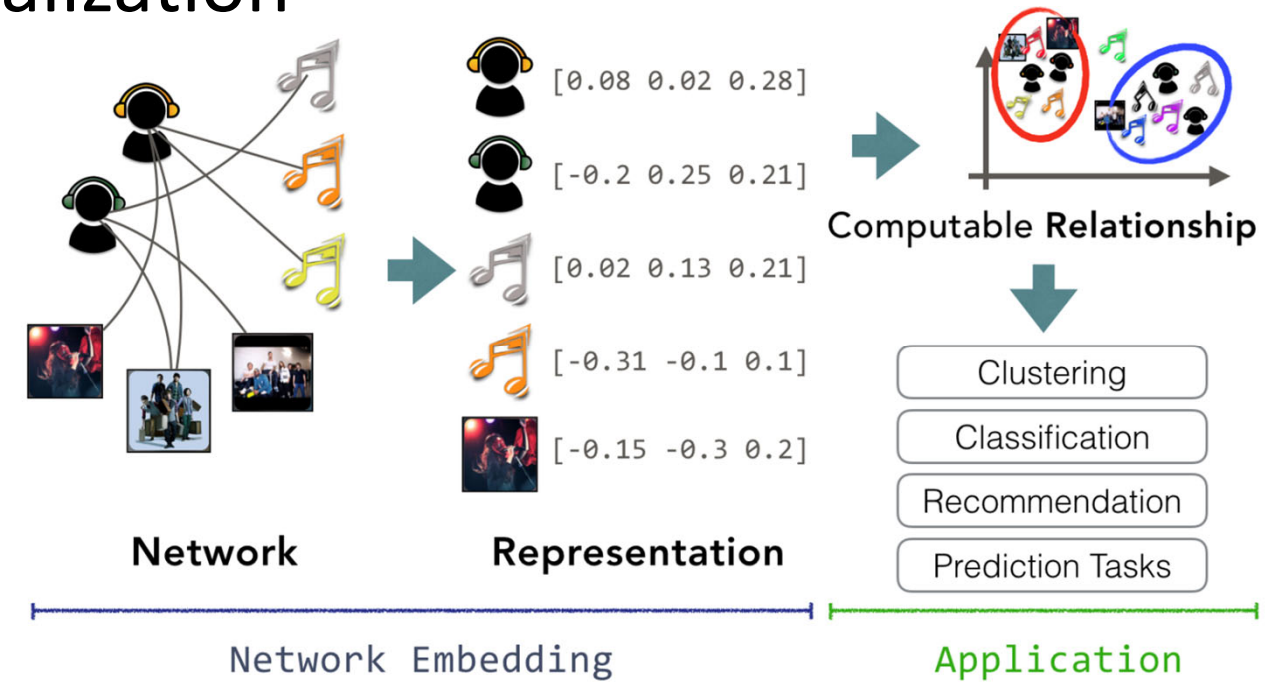
Traditional network analysis vs network embedding



Peng Cui et al., "A Survey on Network Embedding"
IEEE Transactions on Knowledge and Data Engineering, 2017
https://arxiv.org/abs/1711.08752

Network embedding

- network -> vector
  - Similar vertices should be located closer
- beyond visualization

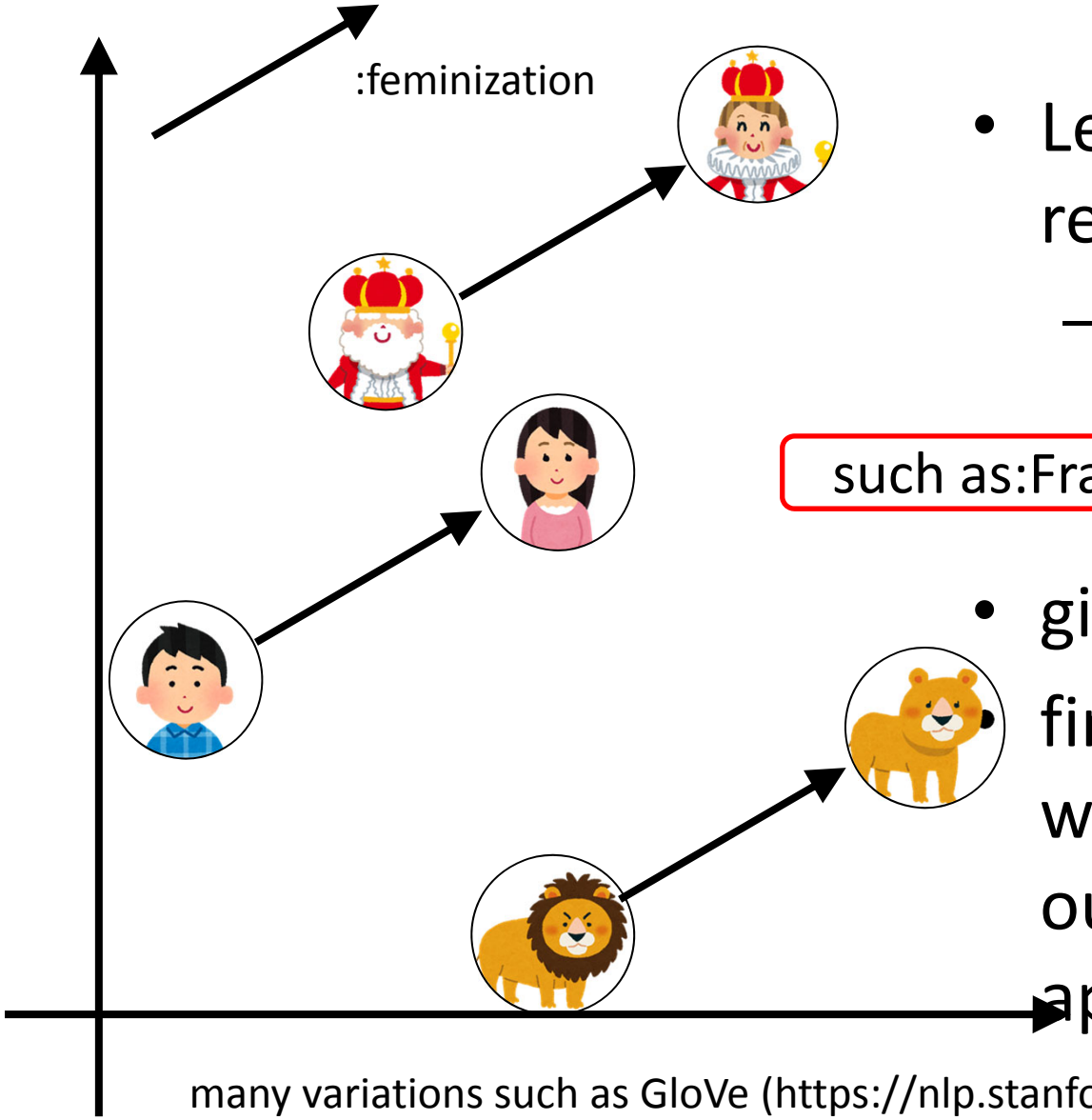


From “awesome network embedding”
https://github.com/chihming/awesome-network-embedding

Why vectors?

- Pros:
  - Many ML tools are available for vectors
  - Attribute information of vertices (such as age, gender, affiliation, …) can be combined with structural information
- Cons:
  - “Goodness” of network embedding is still controversial

Word2vec (Skip-gram)



- Learns vector representation of words
  - addition / subtraction of the meaning is possible !

such as:France – Paris + Japan = Tokyo

not network

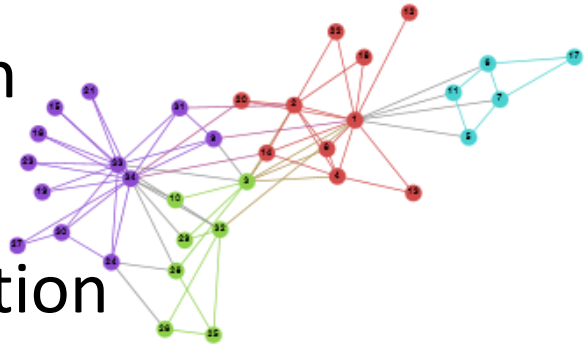
- given: sentences

  find: DNN that accepts words as its input and outputs vector of co-appearing words

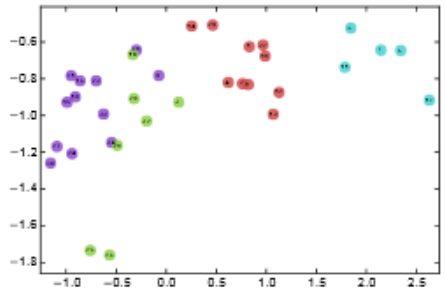
many variations such as GloVe (https://nlp.stanford.edu/projects/glove/)

DeepWalk [Perozzi, KDD2014]

- A network is represented as the sequences of random walks on it
- Then the sequences are given to word2vec (Skip-gram) in order to get vectors
- Then the vectors are used for:
  - Node classification
  - Link prediction
  - Community detection



(a) Input: Karate Graph

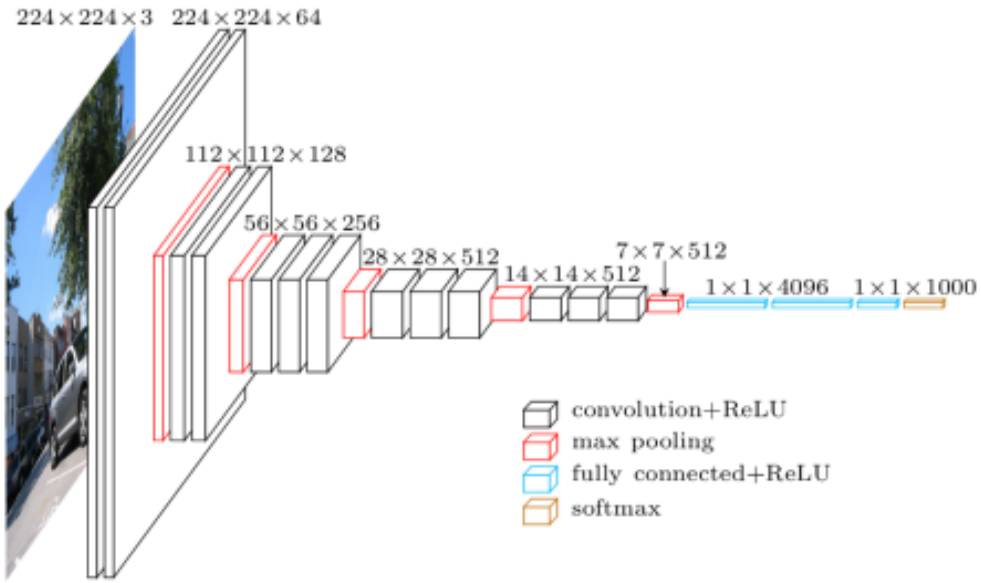


(b) Output: Representation

Pros & cons of network embedding

- Very powerful for some tasks
  - Classification
  - Link prediction
  - ML tasks in combination with node attributes
- Still not perfect:
  - embedding non-simple networks (dynamic, heterogeneous, signed, multiplex, …) is not easy
  - Not good for some tasks (computing diameter, propagating info., searching, …)
  - Comprehensibility (hard to visualize high-dimensional vectors)

Deep Neural Networks (DNNs)



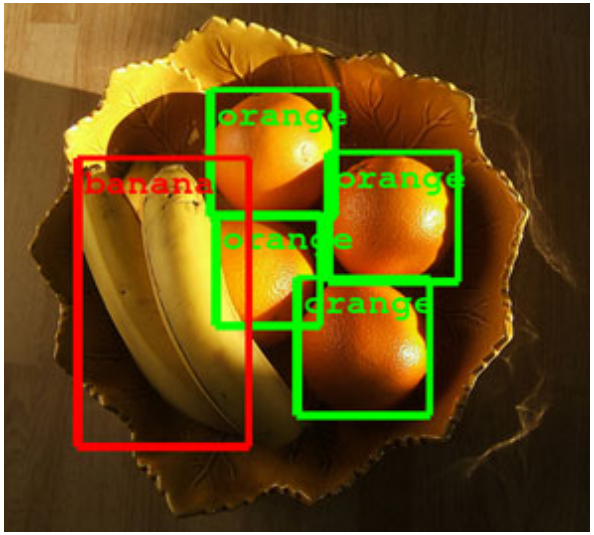


Image recognition

Voice recognition

Natural language processing

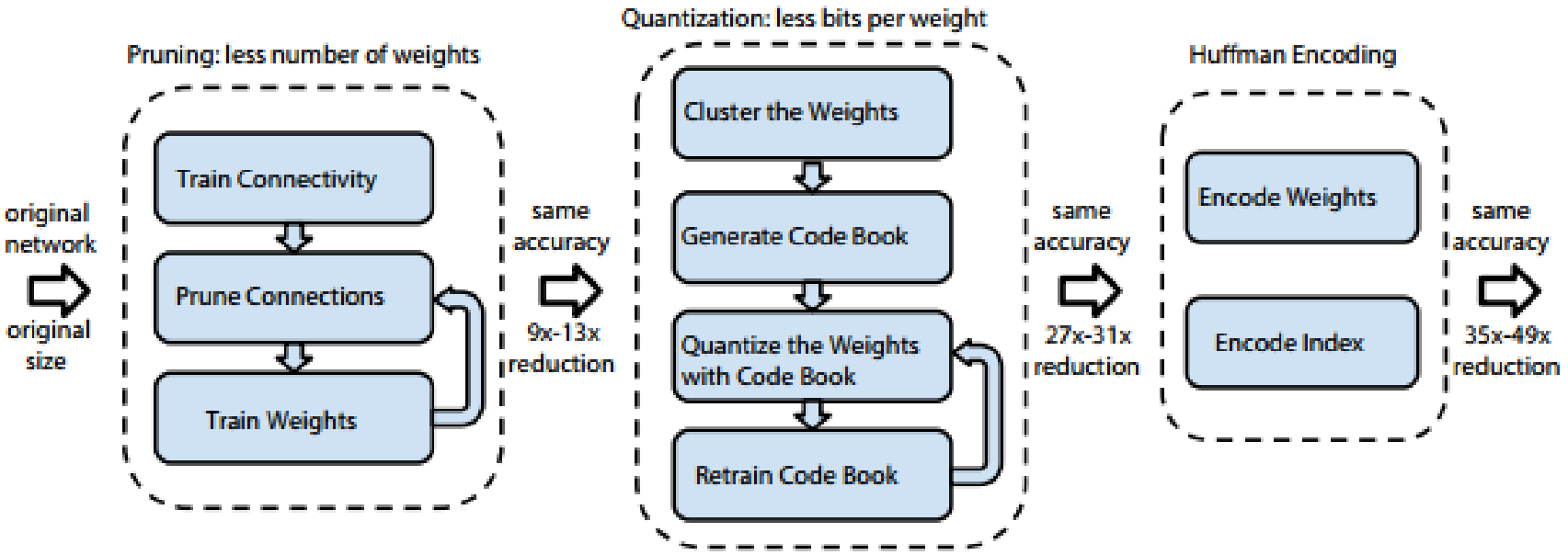
DNN models consume too much storage

- Size of commonly used DNNs
  - AlexNet 240MB
  - VGG 16 552MB
  - Inception V3 109MB
- Running models on the cloud has its own disadvantages
  - Network latency
  - Privacy

DNN Compression

- Can we achieve the same accuracy with smaller models?
- There are several approaches to obtain smaller models
  - Compressing pre-trained networks
    - Deep Compression (Han et al., 2016)
  - Designing of compact models
    - Squeezenet (Iandola et al, 2016)
    - MobileNets (Howard et al., 2017)

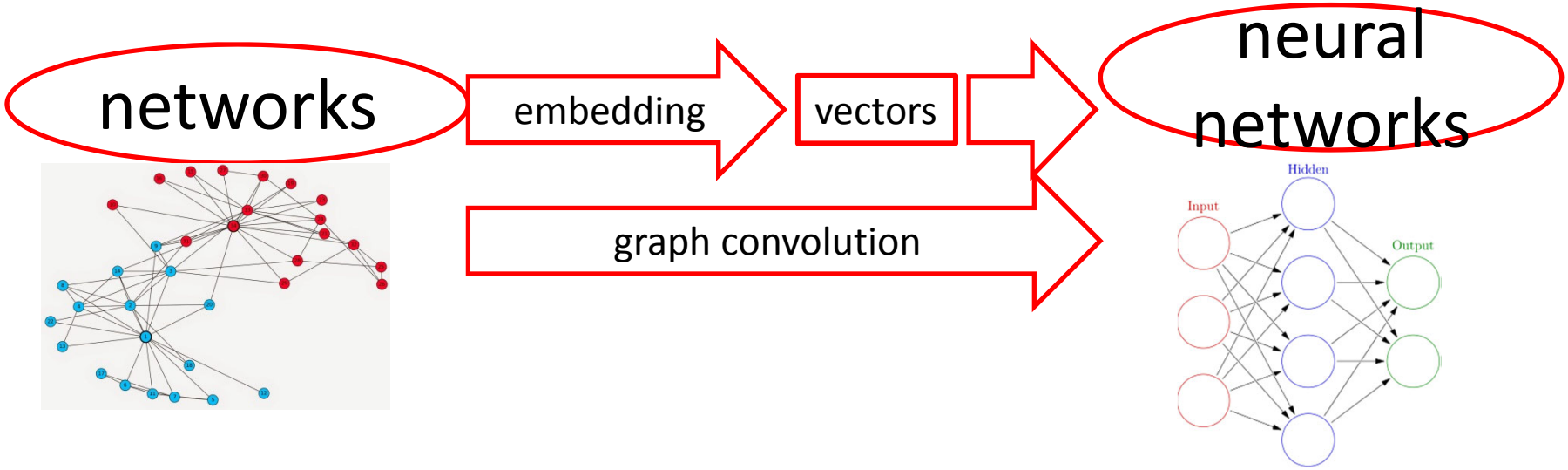
Deep Compression (Han et a., ICLR 2016)



- Commonly referred as state-of-the-art compression
- Requires specific custom hardware to leverage inferencing (Han et al., ISCA 2016)

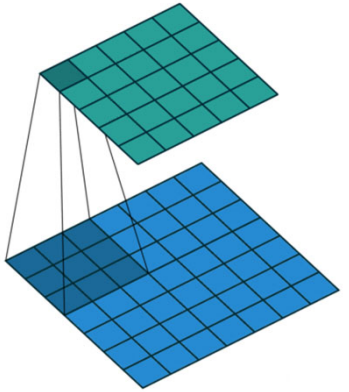
Networks as the input to Neural Networks

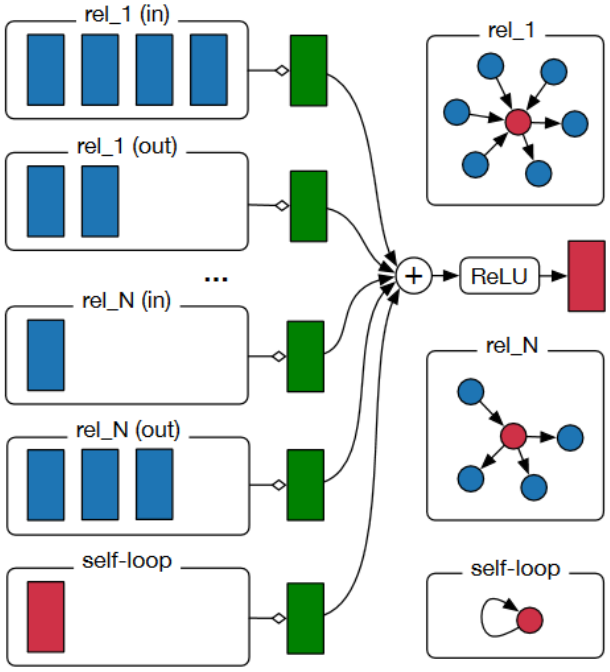
- NN basically accepts vectors as its input
  - network embedding
  - graph convolution



Convolution on graph

- Convolution of images
  - aggregating pixels in a filter
- Convolution of graphs
  - (Graph Fourier transform)
  - aggregating neighbors of all rels
- Application
  - classification / link prediction
  - For structured data (such as chemical compounds)

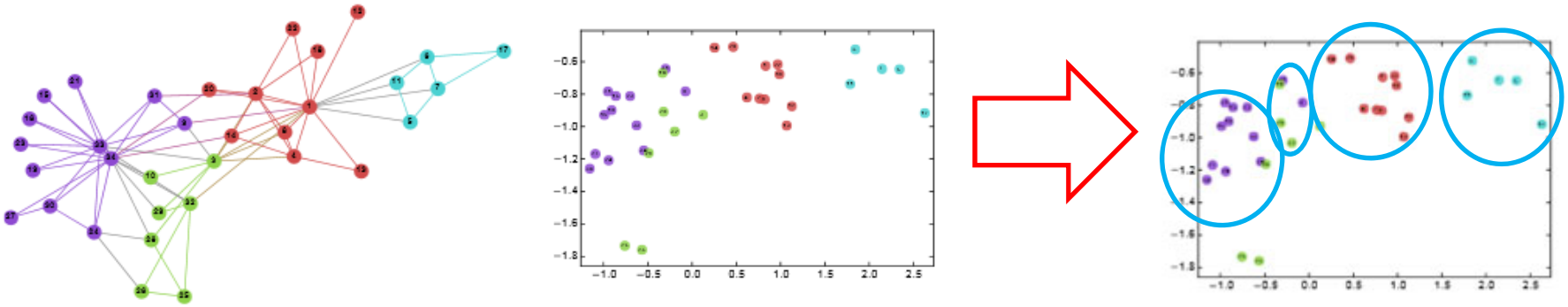




"Modeling Relational Data with Graph Convolutional Networks"
https://arxiv.org/abs/1703.06103

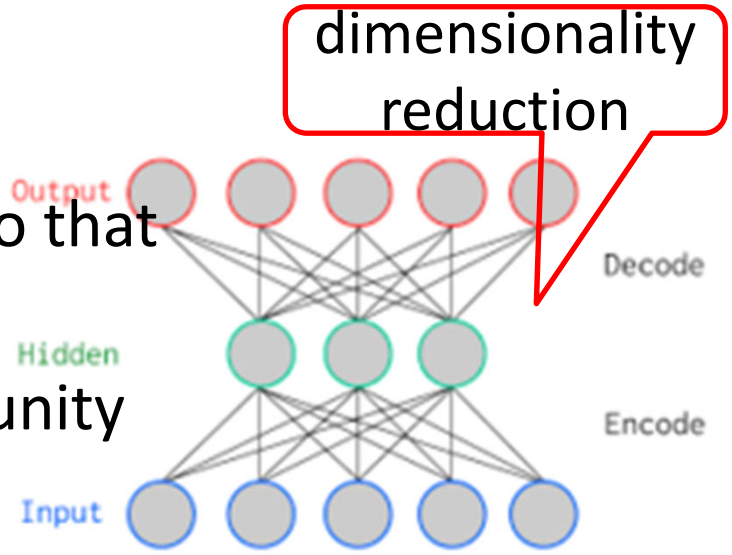
Community detection

- Based on vector representation (obtained from DeepWalk, LINE, …)



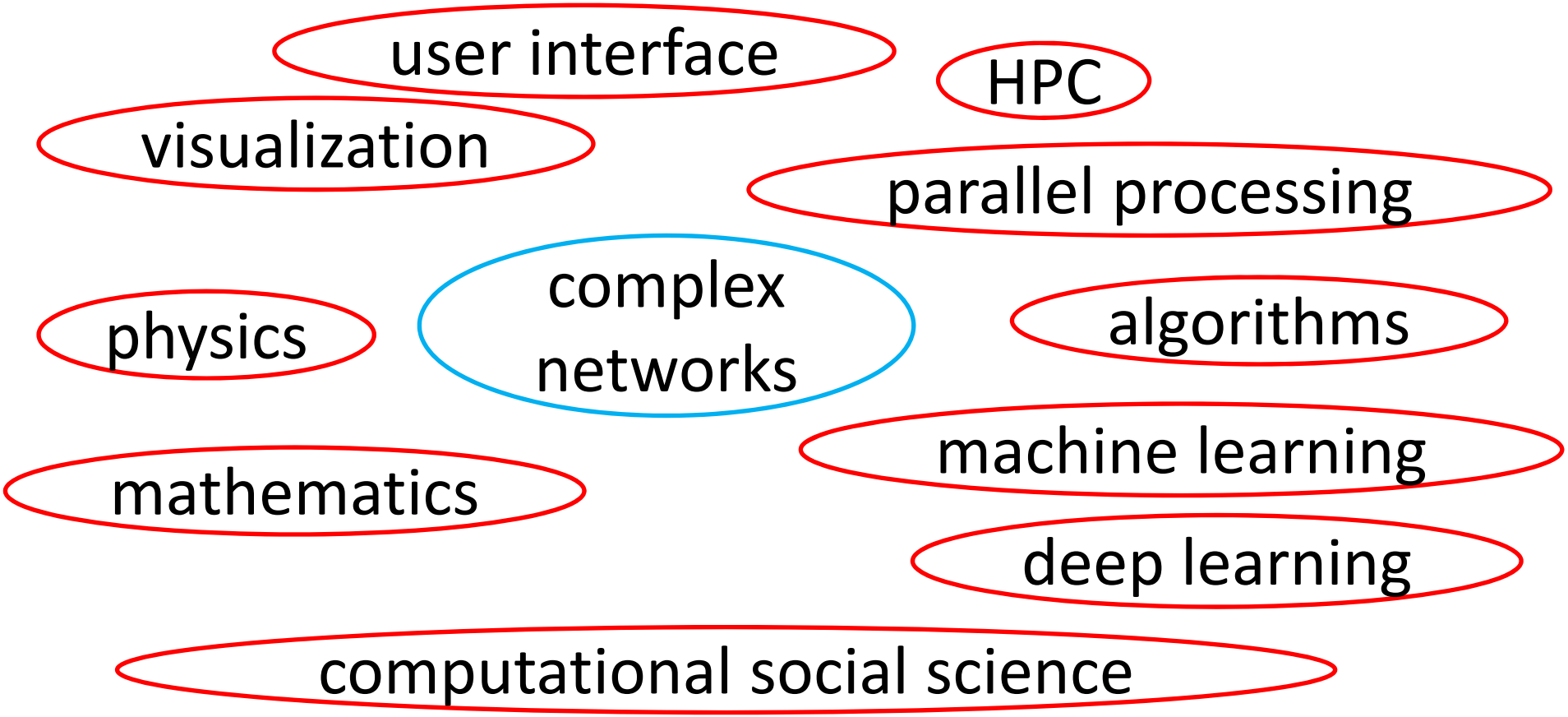
(a) Input: Karate Graph (b) Output: Representation

- Based on autoencoder
  - encoder & decoder are trained so that input & output will be the same
  - Hidden layer are used for community detection



Some references

- SNAP: Stanford Network Analysis Project
  - http://snap.stanford.edu/
- Network Repository
  - http://networkrepository.com/
- Awesome network analysis
  - https://github.com/briatte/awesome-network-analysis
- Social Network Analysis with Python and NetworkX
  - https://pydata.org/barcelona2017/schedule/presentation/7/


user interface
HPC
visualization
parallel processing
complex networks
physics
algorithms
machine learning
mathematics
deep learning
computational social science