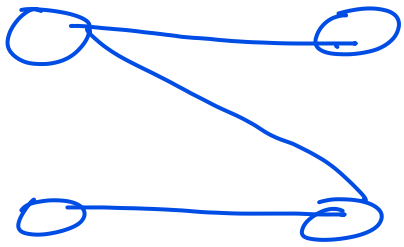
Graph: Graph is general language which describe/analyze relation/interaction amongentities

example:



→ Graph.

Many entity connected with relation

e.g: event graph

① event-graph.
② computer network.
③ ~~Da~~ Disease Pathways
④ Molecules, neurons, road network
⑤ Software → Code graph
⑥ knowledge graph
⑦ Citation network
⑧ social network etc.

Broadly dividing them there will be 2 Categories:

① Naturally Occuring graph
e.g Molecular structure
neurons
Protiens structure etc.

② Graph representation:
code graph
event graph.
knowledge graph etc.

In all this ~~the~~ main question is how

~~do~~ do we take advantage # of relational structure for better prediction?
Complex domain has rich relational structure which can be represented as relational graph.
We need to model this rich relationship structure for better prediction.

If we look at modern deep learning toolbox this there are very effective on grid or sequence kind of data.
Grid: Image
Sequence: Audio, text

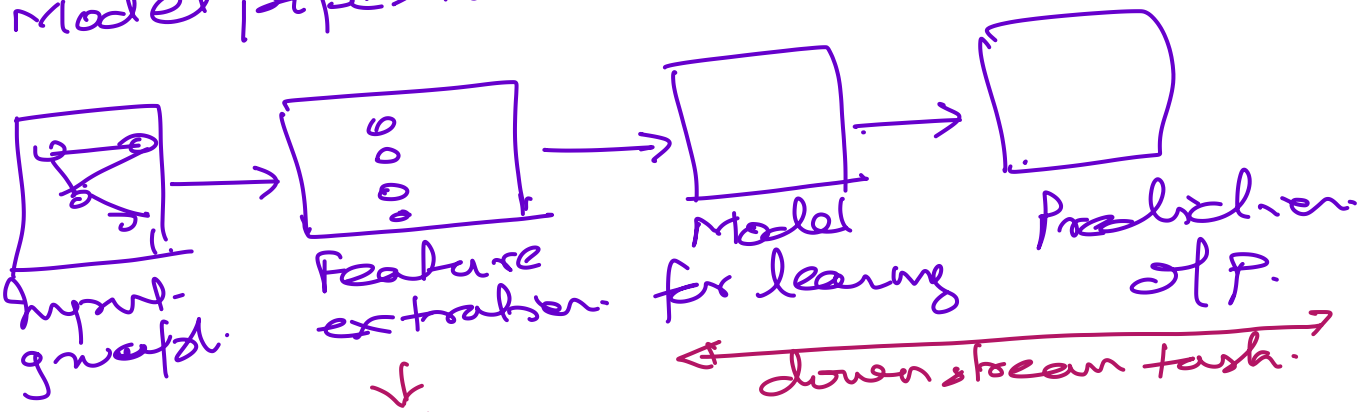
Entites which rich structural information can't be always represented as grid or sequence data. So these DL is not effective
We need more flexible DL which handles structural information ~~more~~ and applicable on these dataset.

Why Graph Deep learning is hard?
(1) No so special locality like grid/sequence.
(2) No starting reference point
(3) Multimodal feature and dynamic in nature.

We need to develop deep learning model which can handle above point.
These model usually perform following task:

1. Node level: node classification
2. Edge level
3. Graph level prediction / Graph generation
4. community / subgraph level task

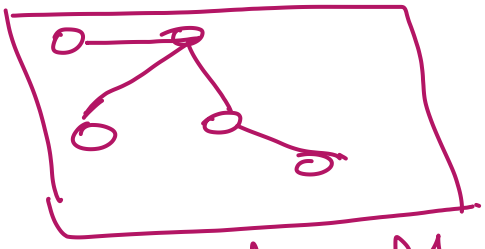
Model pipeline



↓

Automatic feature learning.

what is feature learning ?.



Input graph.

$f: u \rightarrow \mathbb{R}^d$.

d-dimension.

we define function f such that - if we create d-dimensional embedding for node such that similar node embedding will be closer.

There is two ways in which we can create feat node embedding.

1. Manually hand crafted: Traditional ML
2. Automatic learning: Called Representation learning.

Feature Generation:

1. Node level
2. link level
3. Graph level

There is two step in feature extraction:

1. Design the feature
2. Extract feature for training

model performance depends upon feature design. Traditional ML uses hand-designed features

Based on choice of task we need to design d-dimensional feature set.

Objects to generate d-dimensional feature set could be Nodes, edges, set of nodes, entire graphs.

Node level feature generation:

Given $G = (V, E)$

we need to learn a function

$$f: V \rightarrow \mathbb{R}^d$$

function f such that. It will take V(nodes) as input and generate d-dimensional real value vector.

Question: How to design this function:

when we say Node feature then that means how to convert relation/interaction/ structure into feature set. We do not mean node attribute or link attribute.

Node feature design

- Importance based
  - Node degree
  - Node centrality
- structure based
  - Node degree
  - Graphlets.
  - vectors.
  - Cluster coefficient

Node degree based.

Node degree

$f: V \to d(V_i)$ or N_i

$M : X_i \to Y_i$

label.

Y_i

centrality based: —
- Betweeness centrol.
- closeness
- clustering coefficient.

$M : X_i \to Y_i$