saama

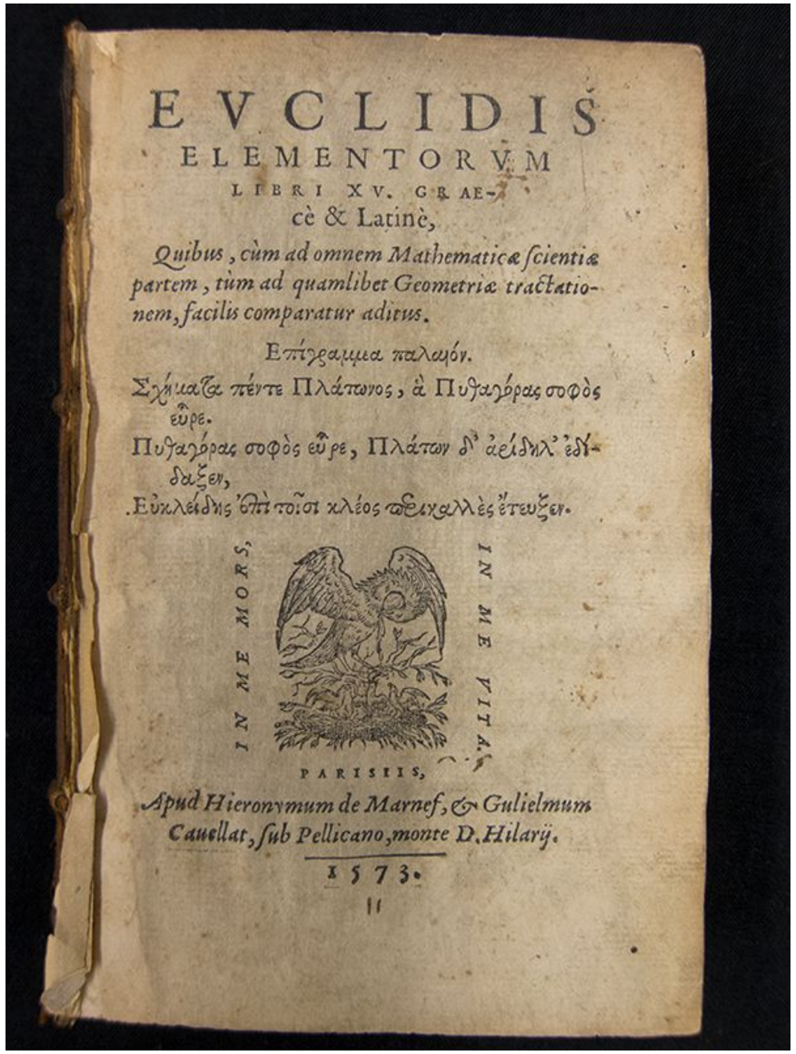
Geometric Deep Learning

Geometric Deep Learning is not just about using Deep Learning methods for Geometrical Datasets likes Graphs

Geometry

GEOMETRON (Greek)
GEO (Earth) + Metron (Measurement)

Part of Philosophy about the Perfection of the Universe
- Pythagoras



EVCLIDIS
ELEMENTORVM
LIBRI XV. GRAE-
cè & Latinè,

Quibus, cùm ad omnem Mathematicæ ſcientiæ partem, tùm ad quamlibet Geometriæ tractationem, facilis comparatur aditus.

Ἐπίγραμμα παλαιόν.

Σχήματα πέντε Πλάτωνος, ἃ Πυθαγόρας σοφὸς εὗρε.
Πυθαγόρας σοφὸς εὗρε, Πλάτων δ' ἀείδηλ' ἐδίδαξεν,
.Εὐκλείδης ἐπὶ τοῖσι κλέος περικαλλὲς ἔτευξεν.

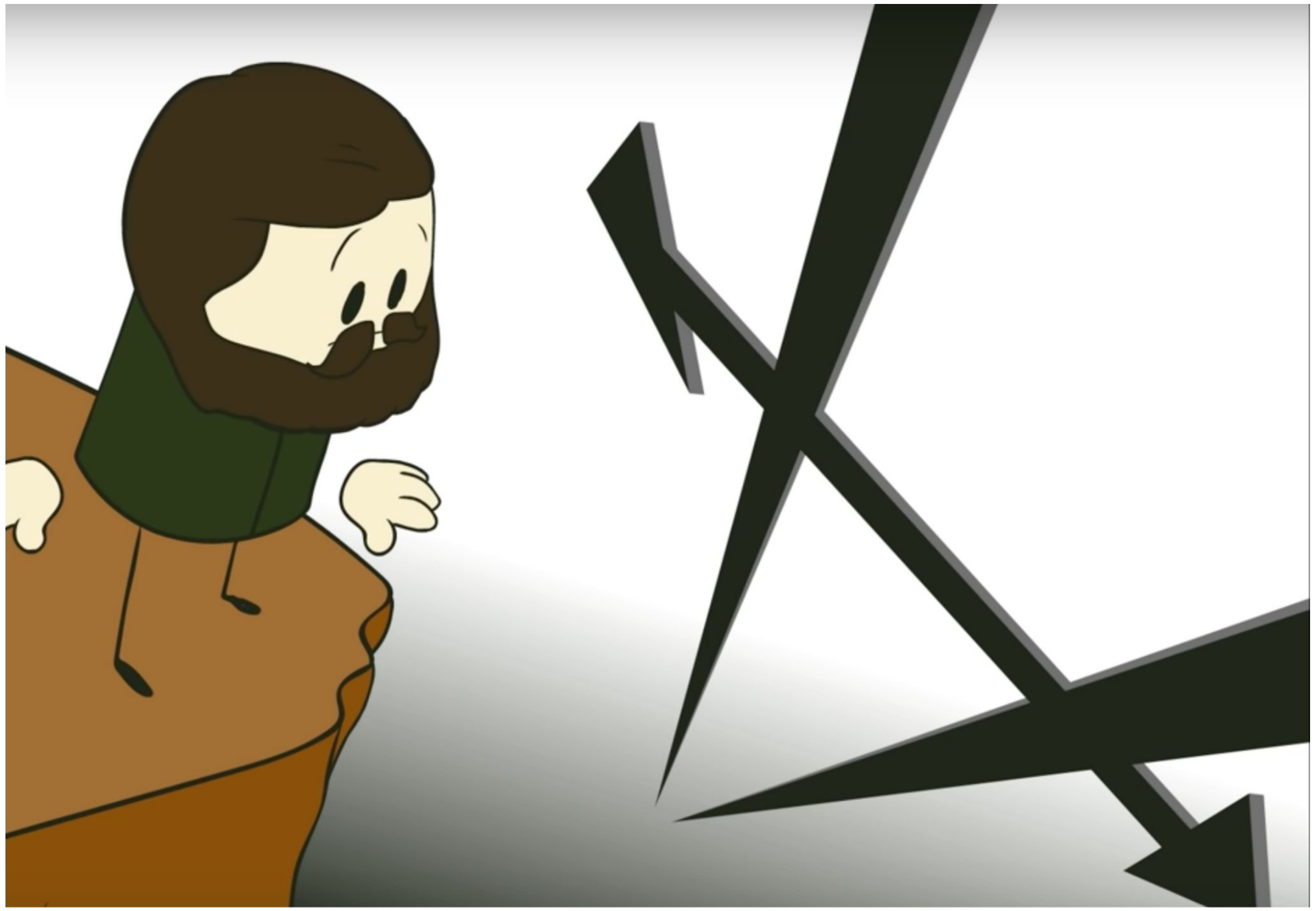
IN ME MORS, IN ME VITA.

PARISIIS,
Apud Hieronymum de Marnef, & Gulielmum Cauellat, ſub Pellicano, monte D. Hilarij.
1573.

This is the Beginning of Both Modern Geometry and Deep Learning

EUCLIDEAN GEOMETRY

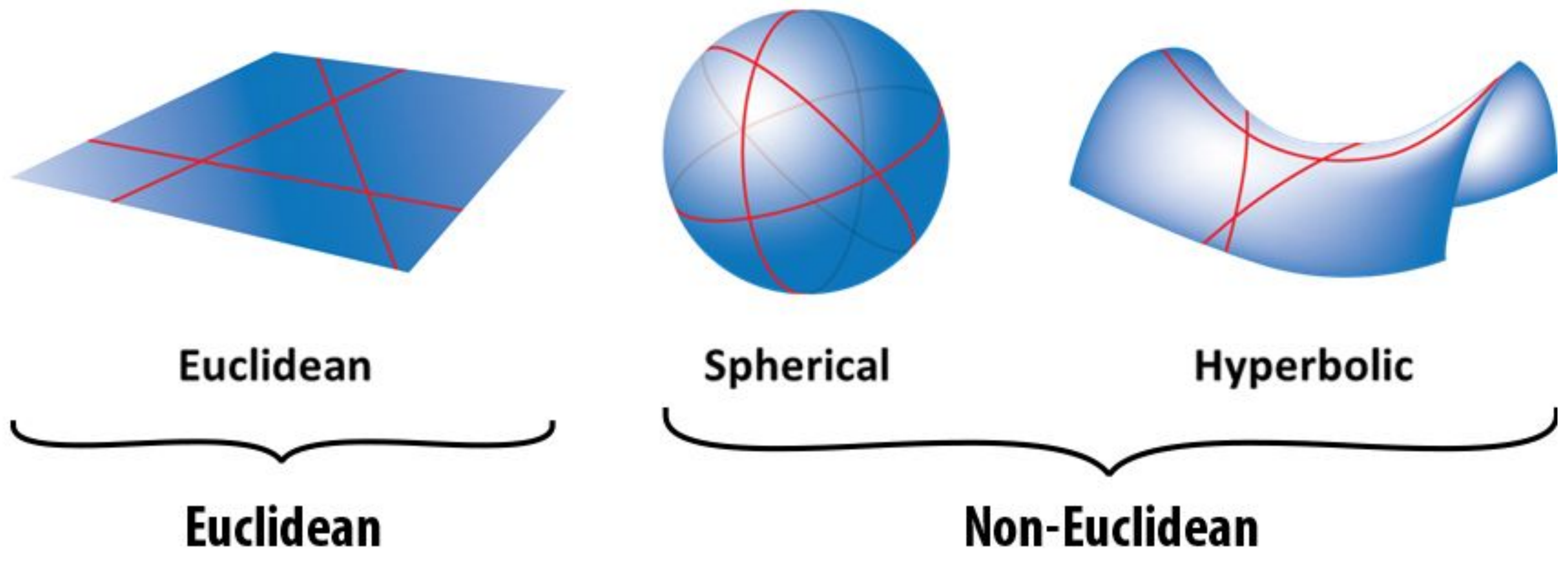
- Plane and solid geometry (2-D)
- 5 Axioms
- Can't Able to prove the 5th one
- Book named ELEMENTS



Do not try the Parallel in that Way, I know that way all along. I have measured that bottomless night, all the light and all the joy of my life went out there.
-**Farcus Bulyai** to his son (April 4th 1820)

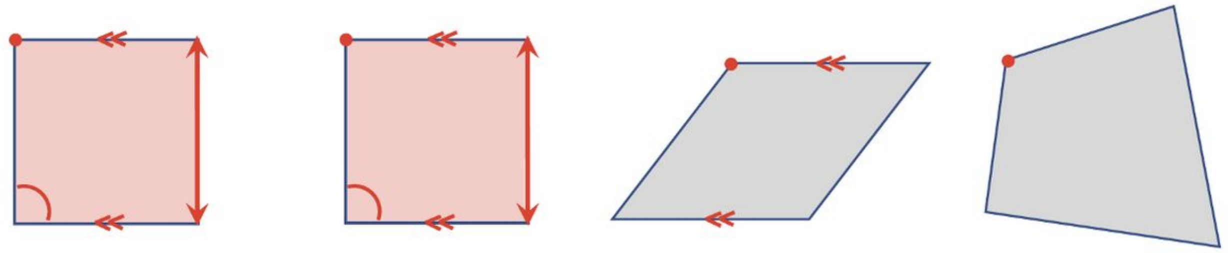
Straight Line might not exists
and
parallel may not function the way you thought they functioned

Non-Euclidean Geometry



ERLANGEN PROGRAM

- Felix Klein
- Geometry
  - Study of Invariants/ Symmentries.
  - The Properties that remains unchanged under some class of transformations
- Different Geometries could be defined by the appropriate choice of symmetry transformations formalized using language of group theory
-

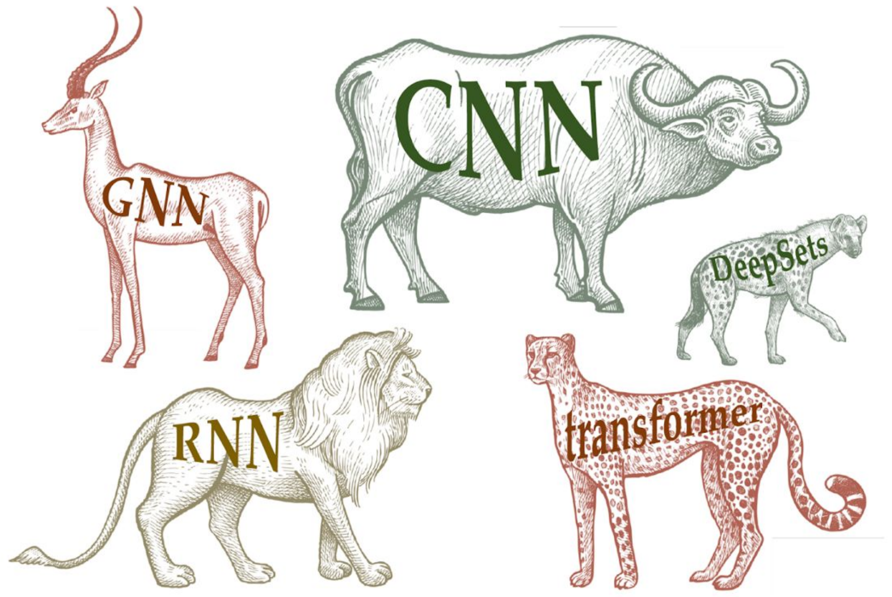


	Euclidean	Affine	Projective
angle	+	−	−
distance	+	−	−
area	+	−	−
parallelism	+	+	−
intersection	+	+	+

Deep Learning - Today

- Several types of Neural Network Architecture for Different types of Datasets with few unifying principle.
- Difficult to understand relationship b/w multiple methods.
- Re-invention of same concepts
- Need Unification

20th Century Zoo of Neural Network Architectures



one of the **Solution**

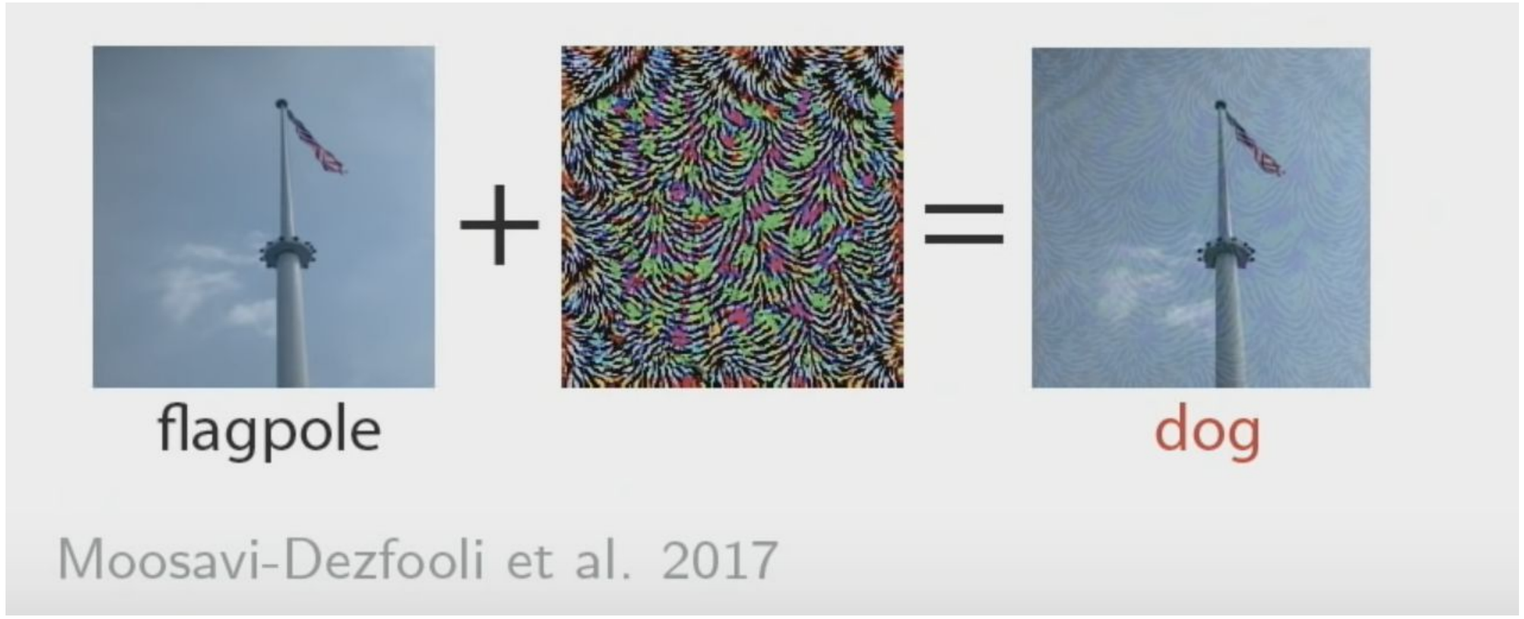
ERLANGEN PROGRAM for DEEP LEARNING

Geometric Deep Learning

- Michael Bronstein
- The Erlangen Programme of ML
- Geometric Deep Learning is an attempt for geometric unification of a broad class of ML problems from the perspectives of symmetry and invariance.
- Mathematical framework to derive NN Architecture
- Have a constructive procedure to build future architecture in principled way

Example

Adding Adversarial Noise on Images makes the Conventional DL Recognition system to Fail



Solution

- PeerNets: Exploiting Peer Wisdom Against Adversarial Attacks
- A novel family of convolutional networks alternating classical Euclidean convolutions with graph convolutions to harness information from a graph of peer samples

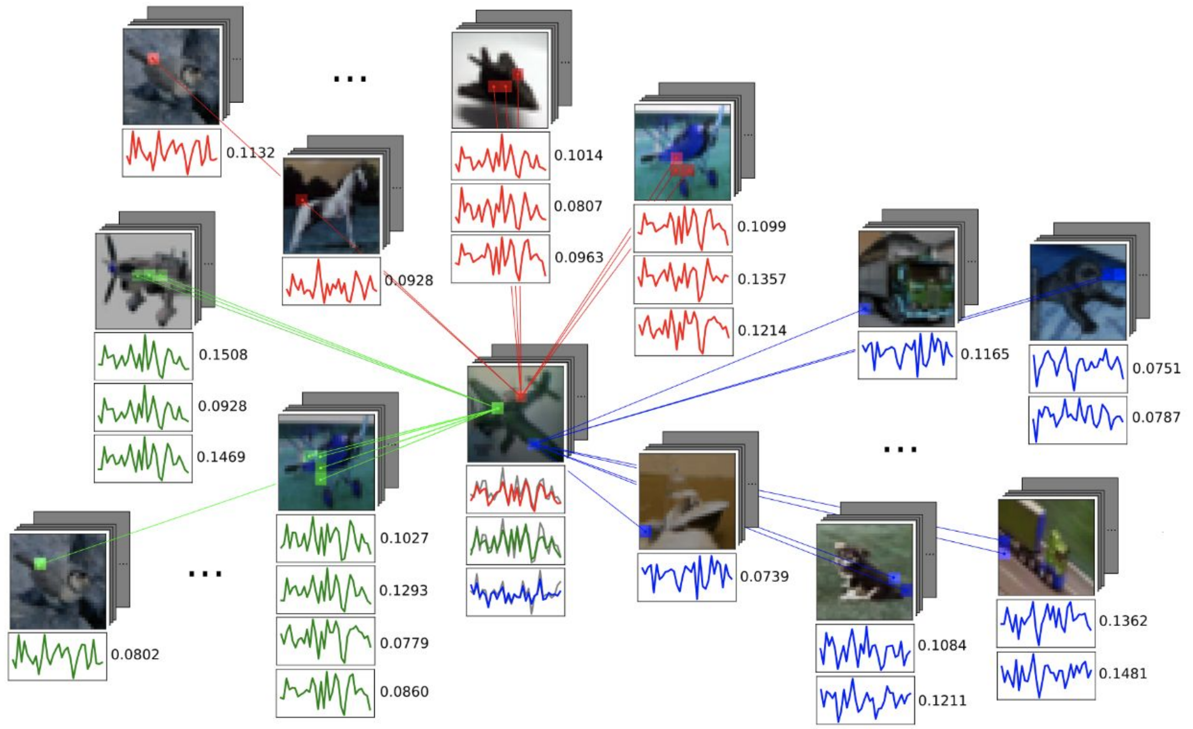


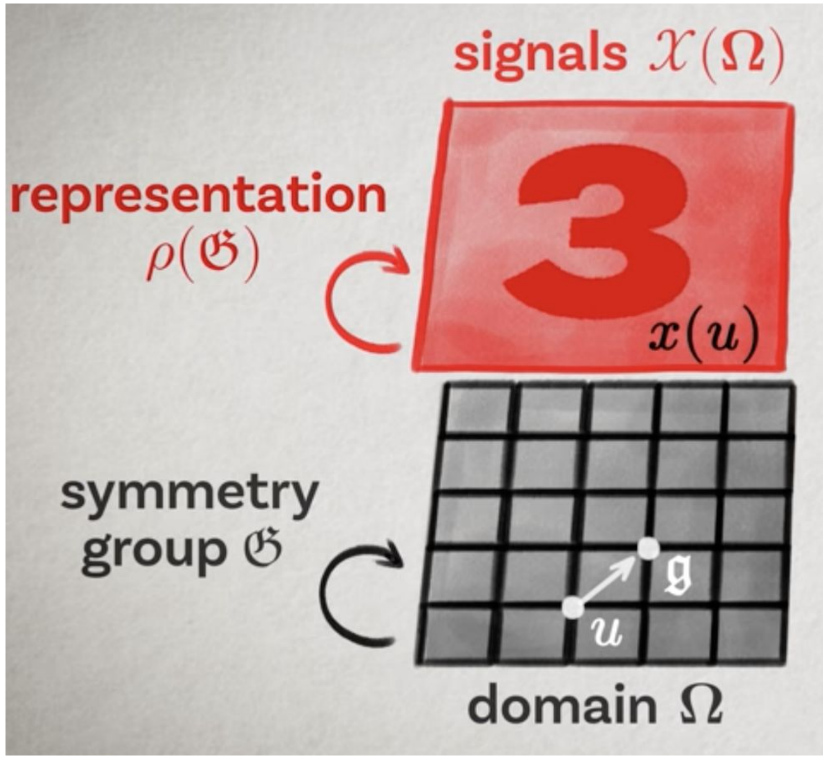
Figure 1: Our Peer Regularization illustrated on three pixels (red, green, blue) of a CIFAR image (center). For each pixel, K nearest neighbors are found in peer images. Plots represent the feature maps in the respective pixels; numbers represent the attention scores.

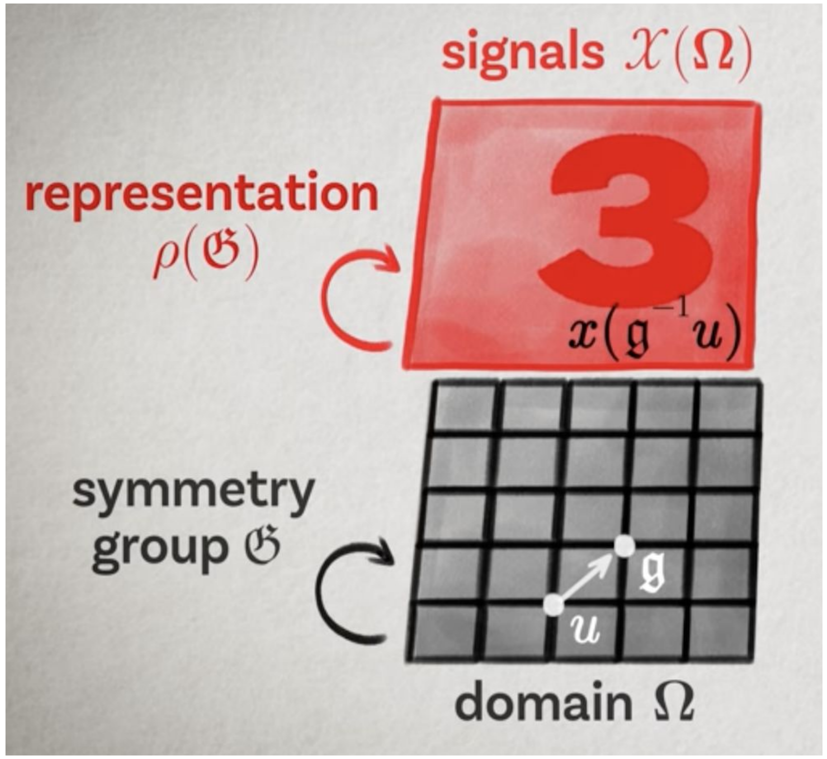
Deep Learning

- Functional Estimation Problem
- Design Rich Functional Classes (Large Datasets with More Computational Resource)
- Low Dimension - Precise mathematical control of Error
- High Dimension - No. Of Sample will grow Exponential

Shift Invariance

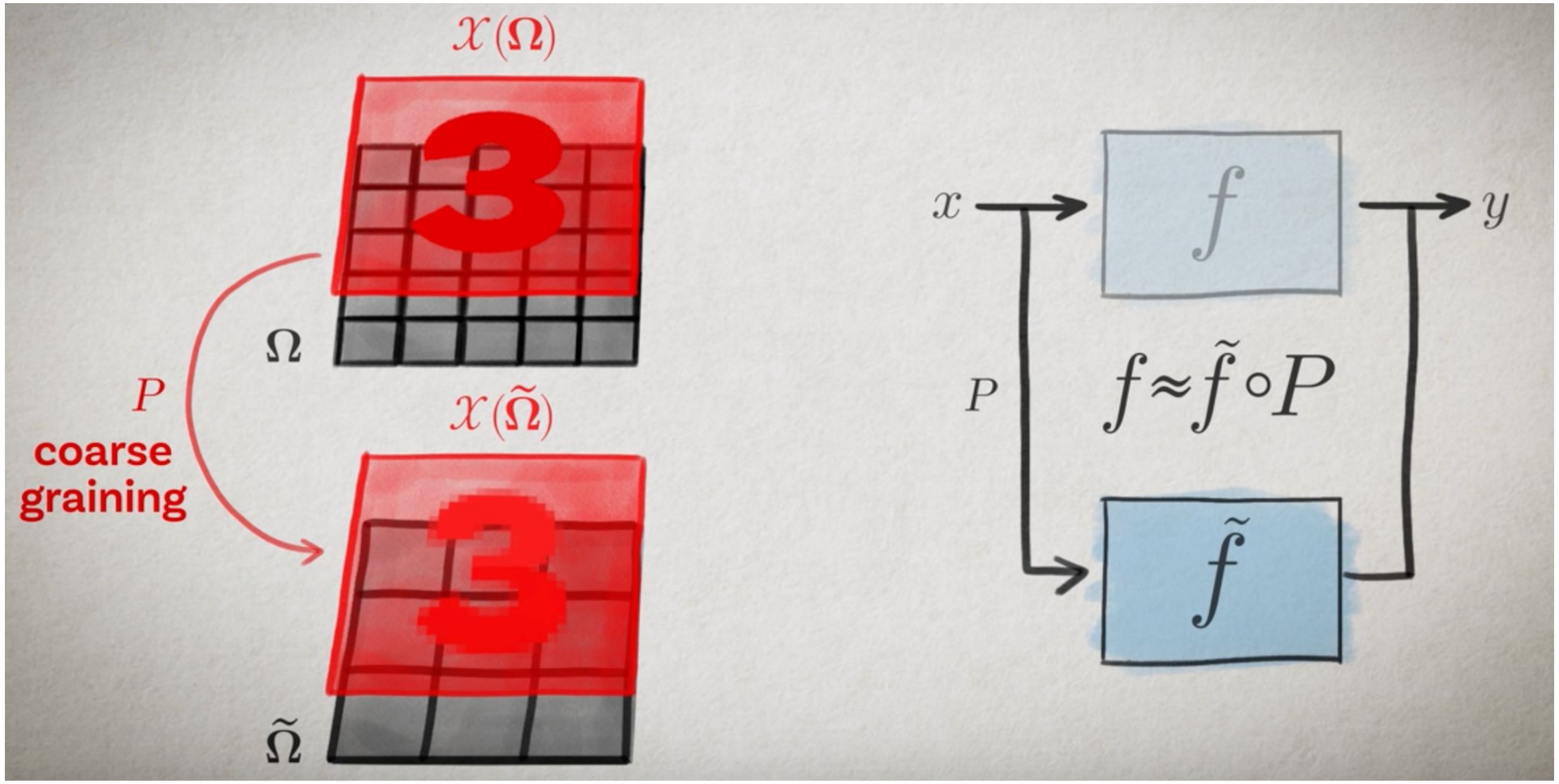
Geometric structure of the Input Geometric Prior (Signal Defined of some Domains)


signals $\mathcal{X}(\Omega)$
representation $\rho(\mathfrak{G})$
$x(u)$
symmetry group $\mathfrak{G}$
$\mathfrak{g}$
u
domain Ω


signals $\mathcal{X}(\Omega)$
representation $\rho(\mathfrak{G})$
$x(\mathfrak{g}^{-1}u)$
symmetry group $\mathfrak{G}$
$\mathfrak{g}$
u
domain Ω

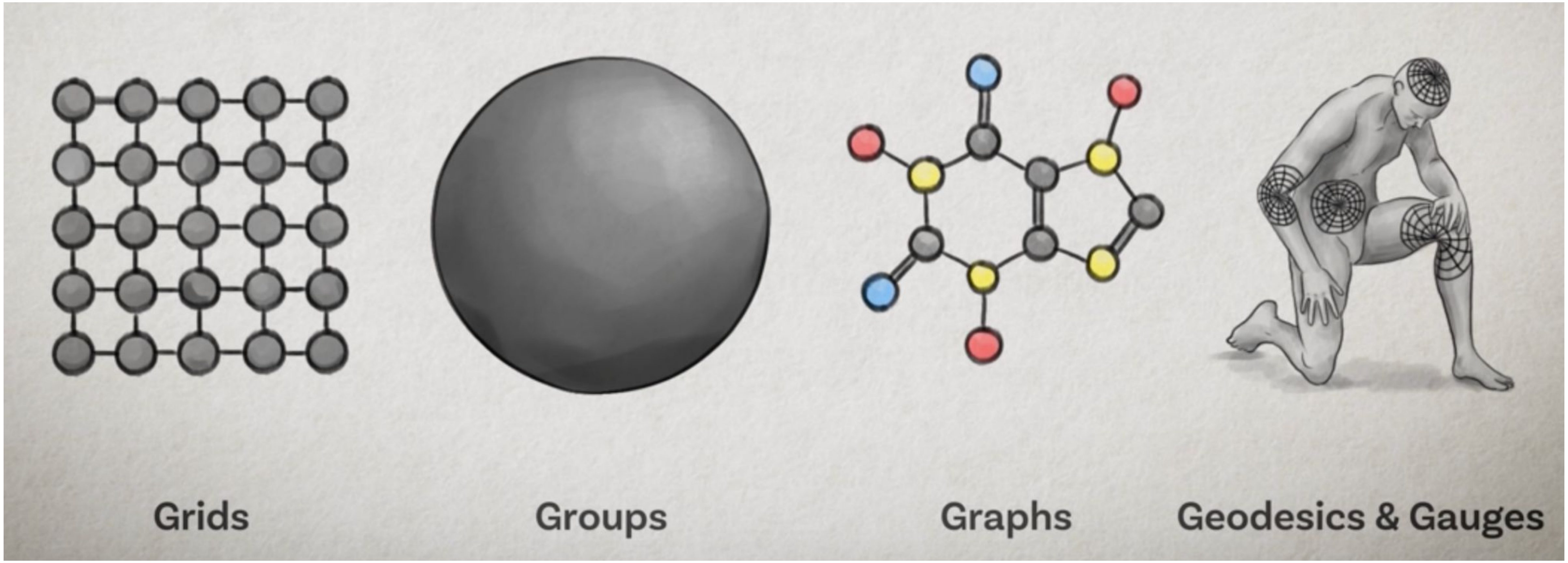
$$\rho(\mathfrak{g})x(u) = x(\mathfrak{g}^{-1}u)$$

Scale Separation



The Above design methods can be applied to Different types of Geometric Structure

5-Gs of Geometric Deep Learning



Graph

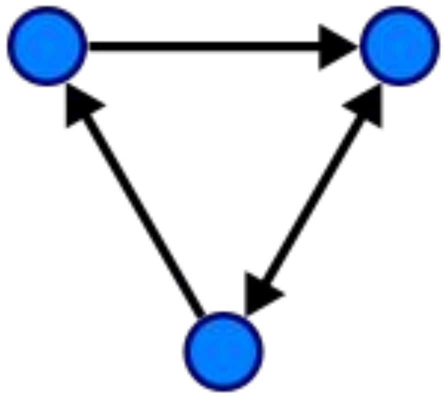
Graph, who?

$$G = (V, E)$$

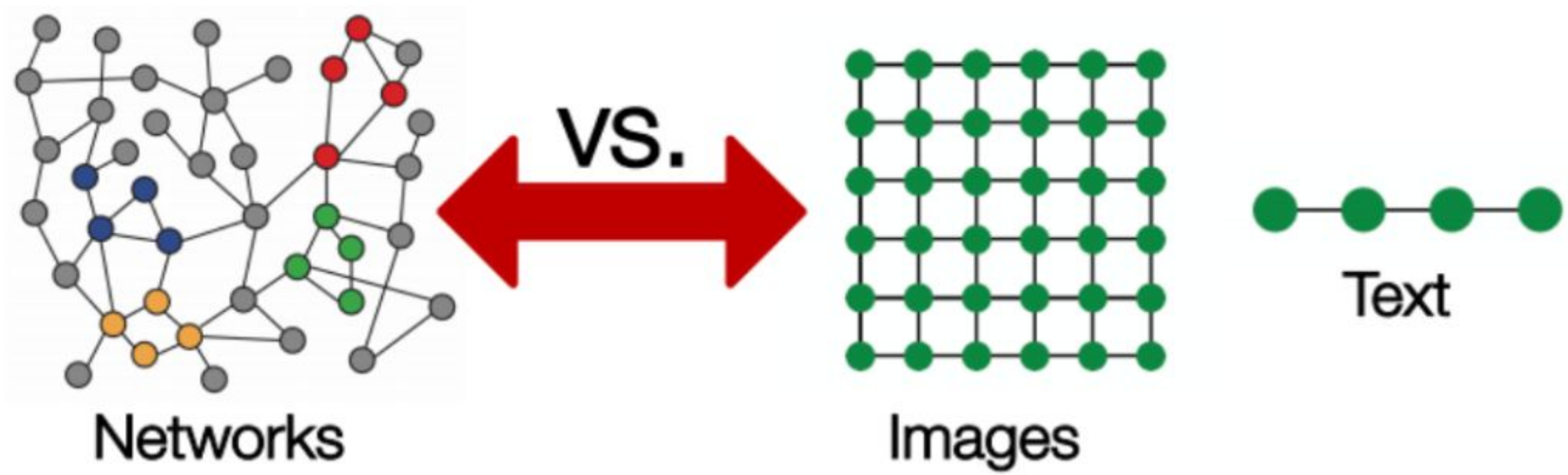
Graph Theory

Graph, who?

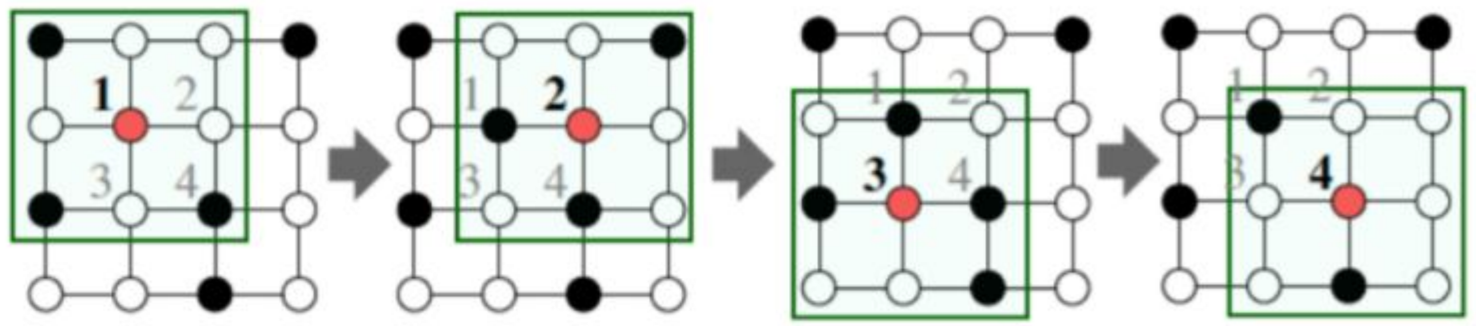
$$G = (V, E)$$

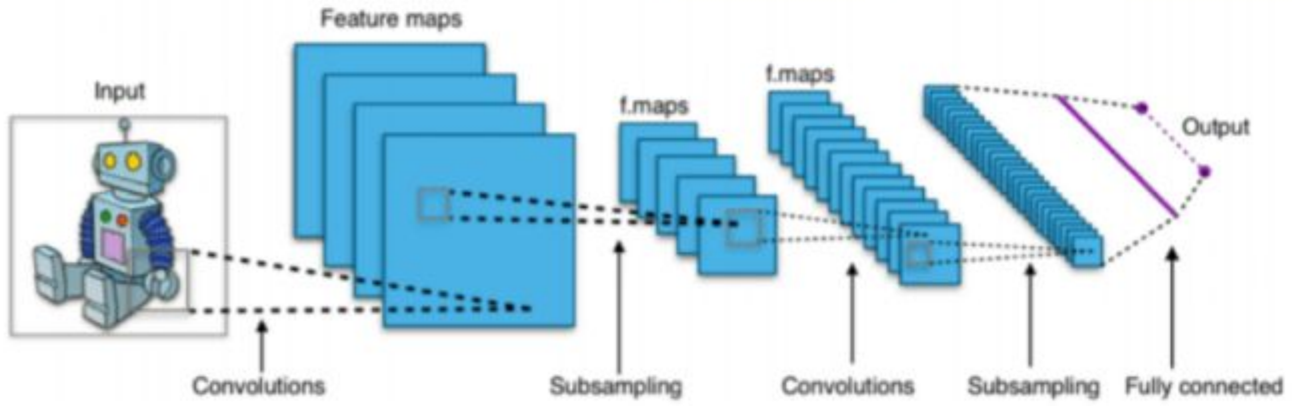


Why is it hard to analyze a graph?



Why do Convolutional Neural Networks (CNNs) fail on graphs?

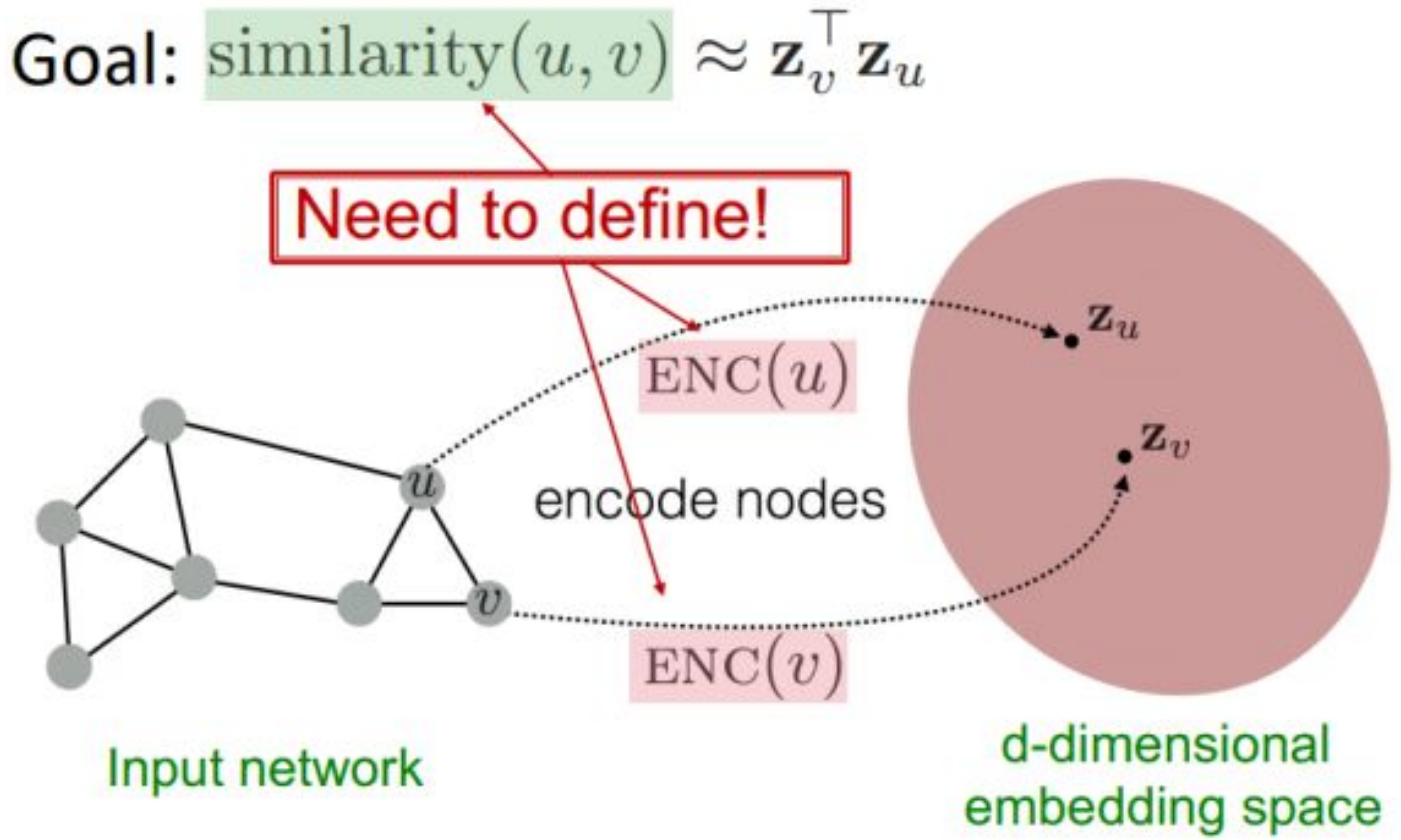




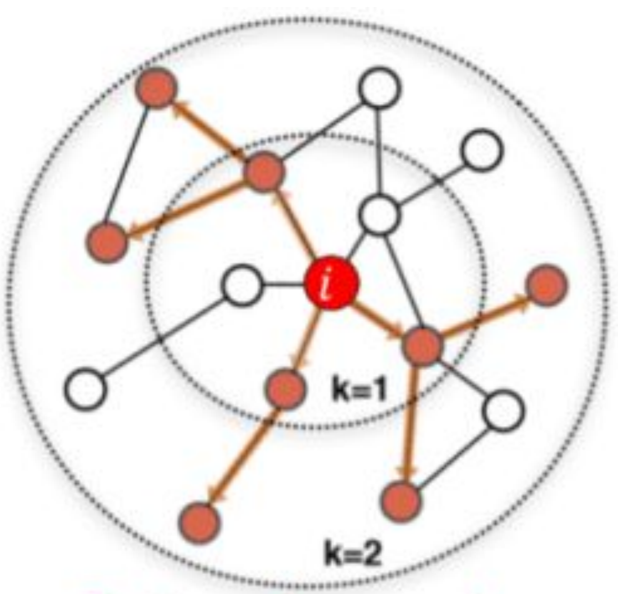
Problems

- Arbitrary size of the graph
- Unfixed node

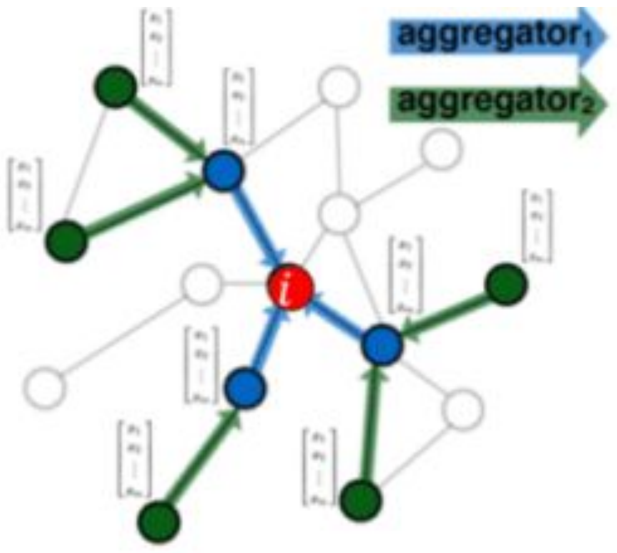
Basics of Deep Learning for graphs



How to come up with the encoder function?



Determine node computation graph

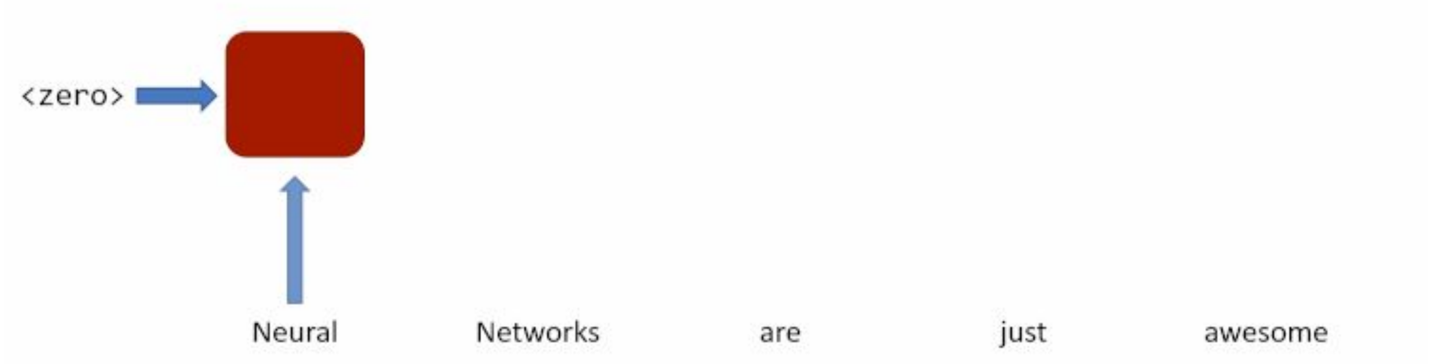


Propagate and transform information

- Locality (local network neighborhoods)
- Aggregate information
- Stacking multiple layers (computation)

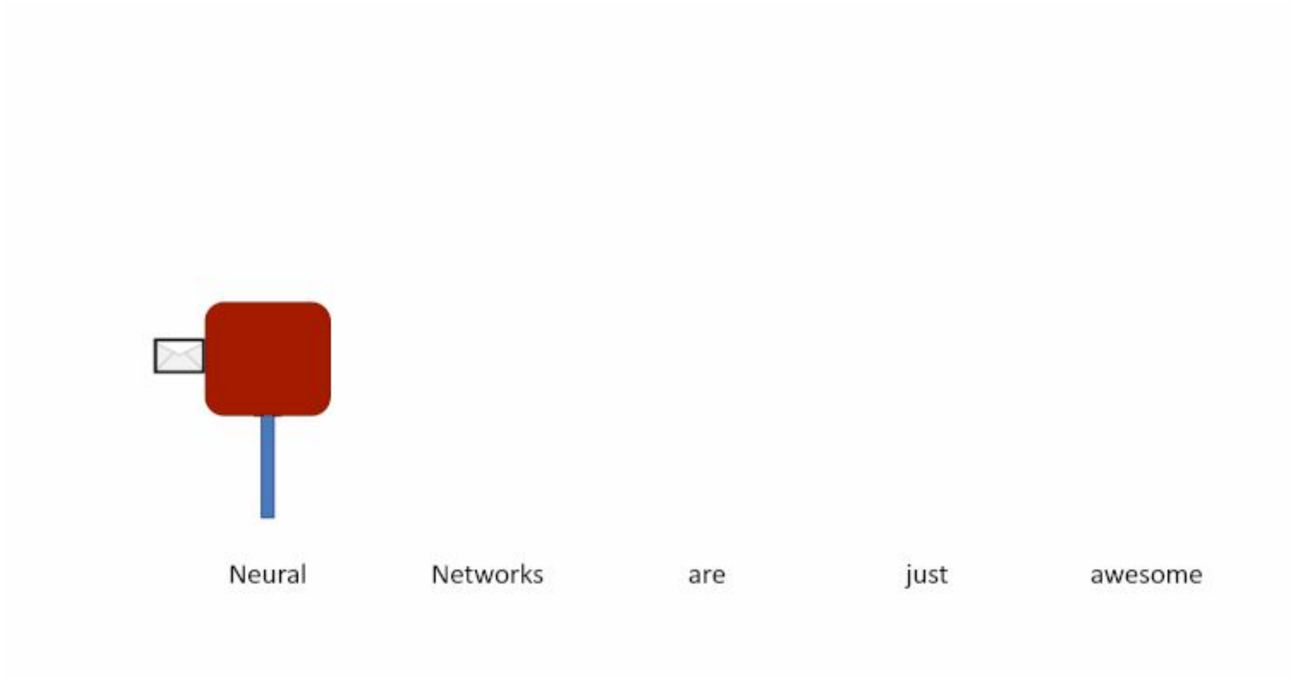
Message Passing

This recurrent simply combines the input — a word for instance — with the output of itself from the previous time-step and passes it through a tanh activation to get the output of the current time-step.



Message Passing

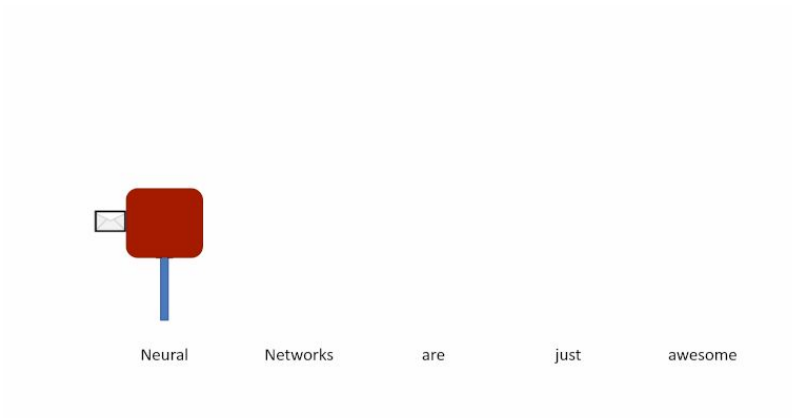
Replace the **state with message**, Each element in the sequence updates the message to incorporate information about itself so that the message reflects the information in the sequence as it is processed.



Carried state as a message that's passed between the sequence elements.

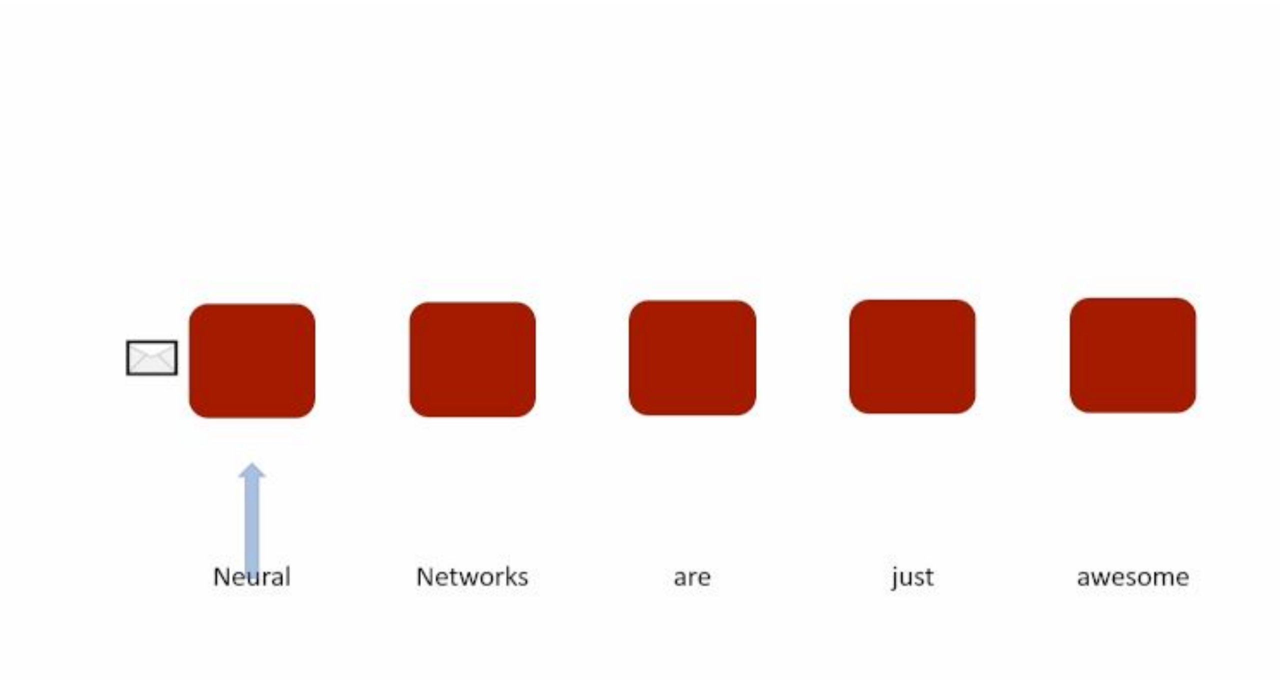
Message Passing

- length of the sequence = Number of time steps
- Essentially, copies of the RNN cell are made over time (unrolling/unfolding), with different inputs at different time steps.
- But graphs can be arbitrarily big and flexible.
- RNN unfold each time step for a message to collect information about it in its entirety. It might seem as though this is a lot of unnecessary work.



Message Passing

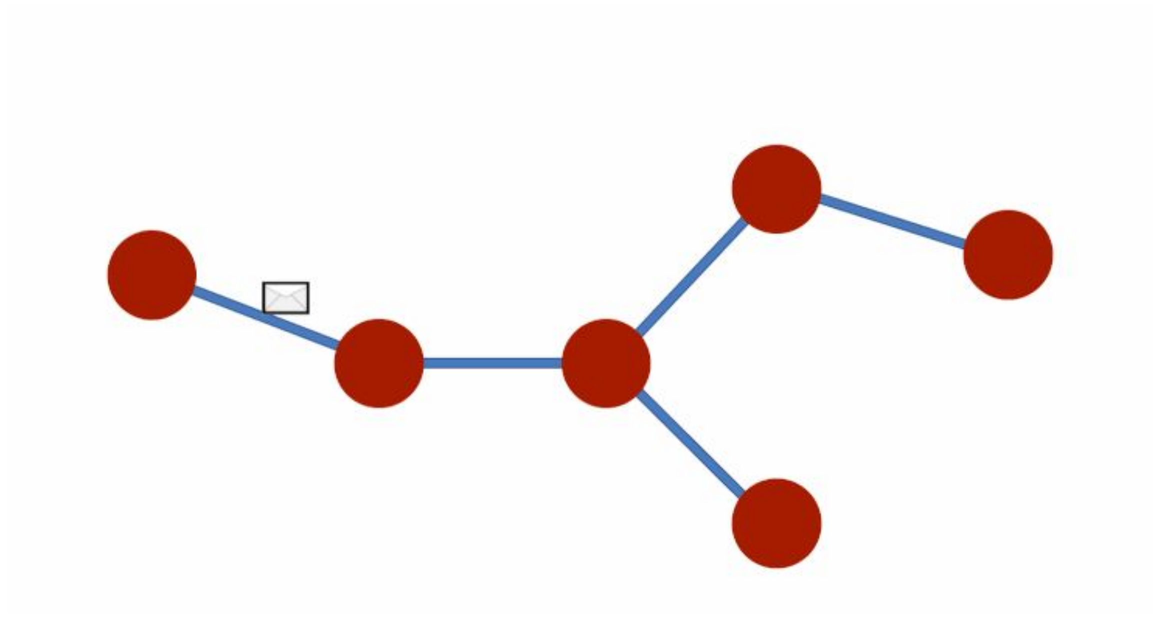
What if an element send a message to the next one on each timestep of the sequence,



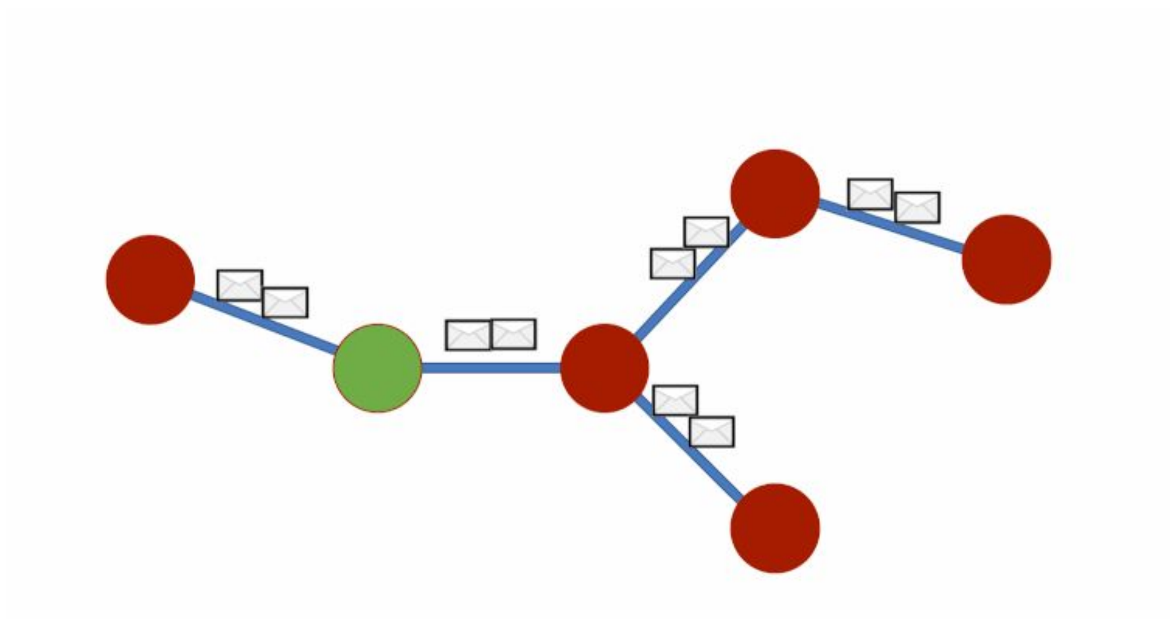
Carried state as a message that's passed between the sequence elements.

Message Passing

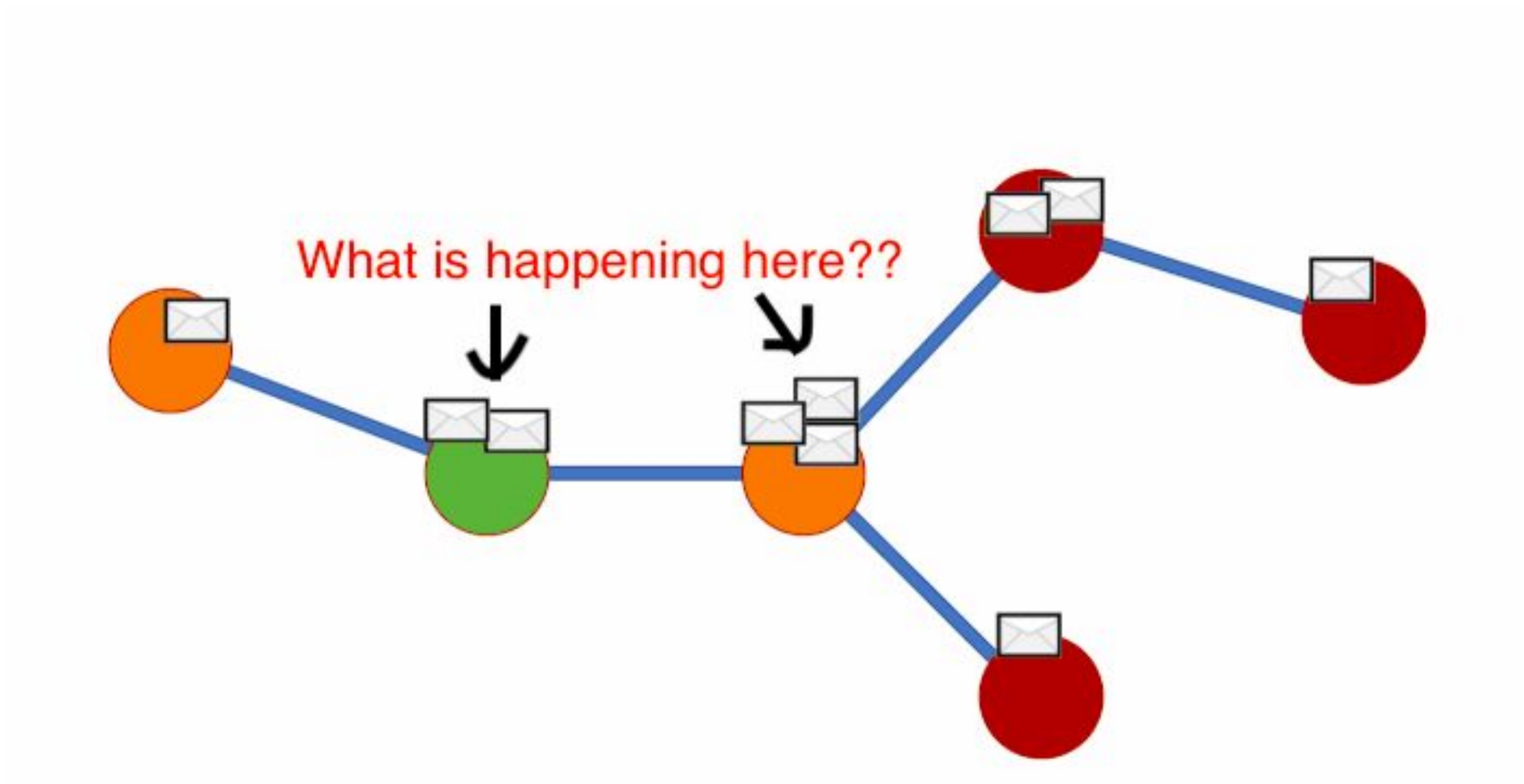
As we applied message passing on sequences, We can apply same idea of message-passing on Graphs



Feature Aggregation

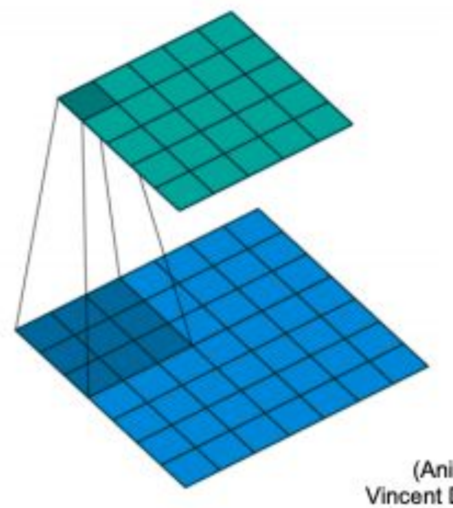


Feature Aggregation

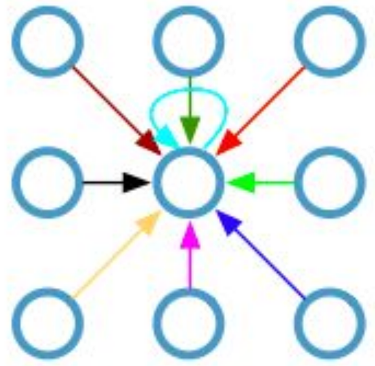


Recap: Convolutional neural networks (on grids)

Single CNN layer with 3x3 filter:

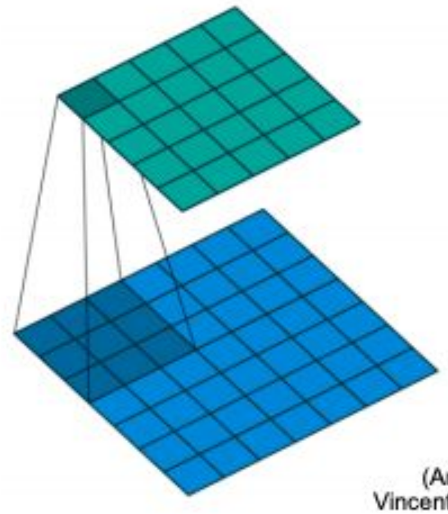


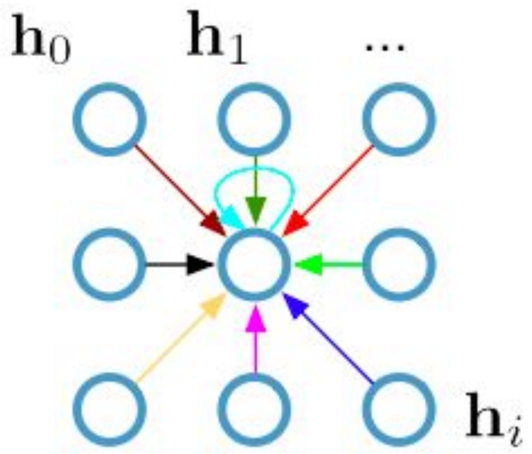
(Animation by Vincent Dumoulin)



Recap: Convolutional neural networks (on grids)

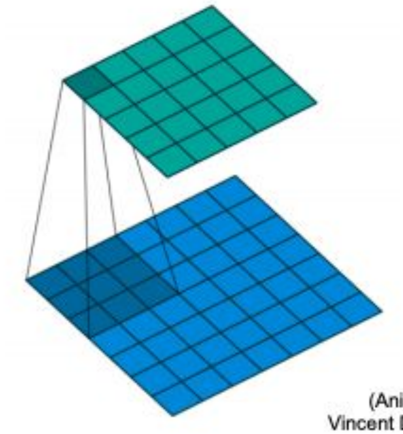
Single CNN layer with 3x3 filter:


(Animation by Vincent Dumoulin)

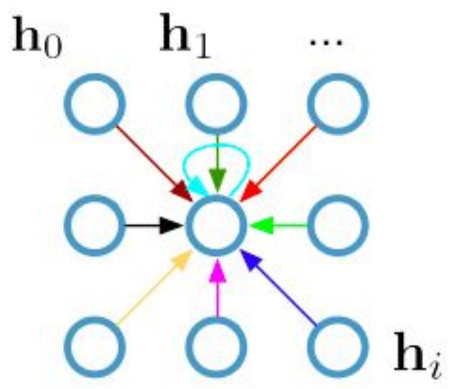


Recap: Convolutional neural networks (on grids)

Single CNN layer with 3x3 filter:



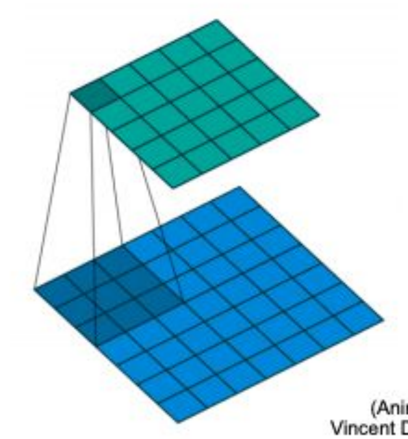
(Animation by Vincent Dumoulin)



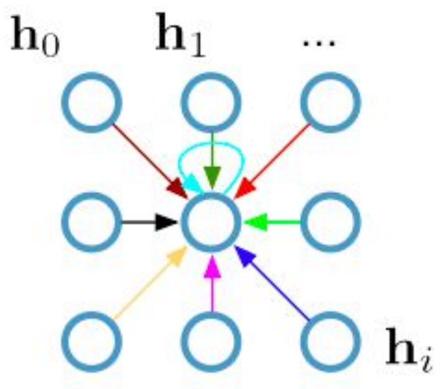
$\mathbf{h}_i \in \mathbb{R}^F$ are (hidden layer) activations of a pixel/node

Recap: Convolutional neural networks (on grids)

Single CNN layer with 3x3 filter:



(Animation by Vincent Dumoulin)



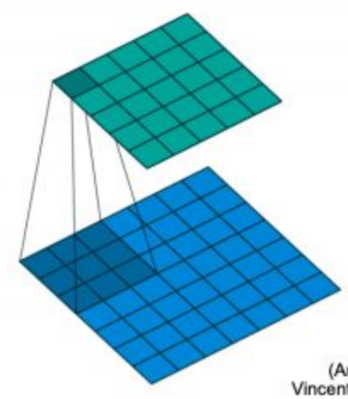
Update for a single pixel:

- Transform messages individually $\mathbf{W}_i \mathbf{h}_i$
- Add everything up $\sum_i \mathbf{W}_i \mathbf{h}_i$

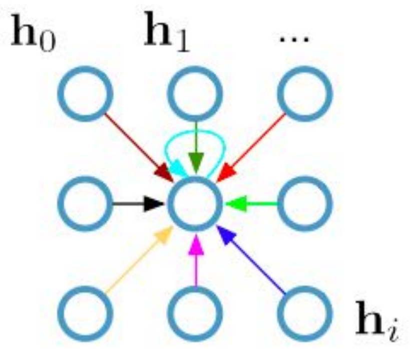
$\mathbf{h}_i \in \mathbb{R}^F$ are (hidden layer) activations of a pixel/node

Recap: Convolutional neural networks (on grids)

Single CNN layer with 3x3 filter:



(Animation by Vincent Dumoulin)



Update for a single pixel:

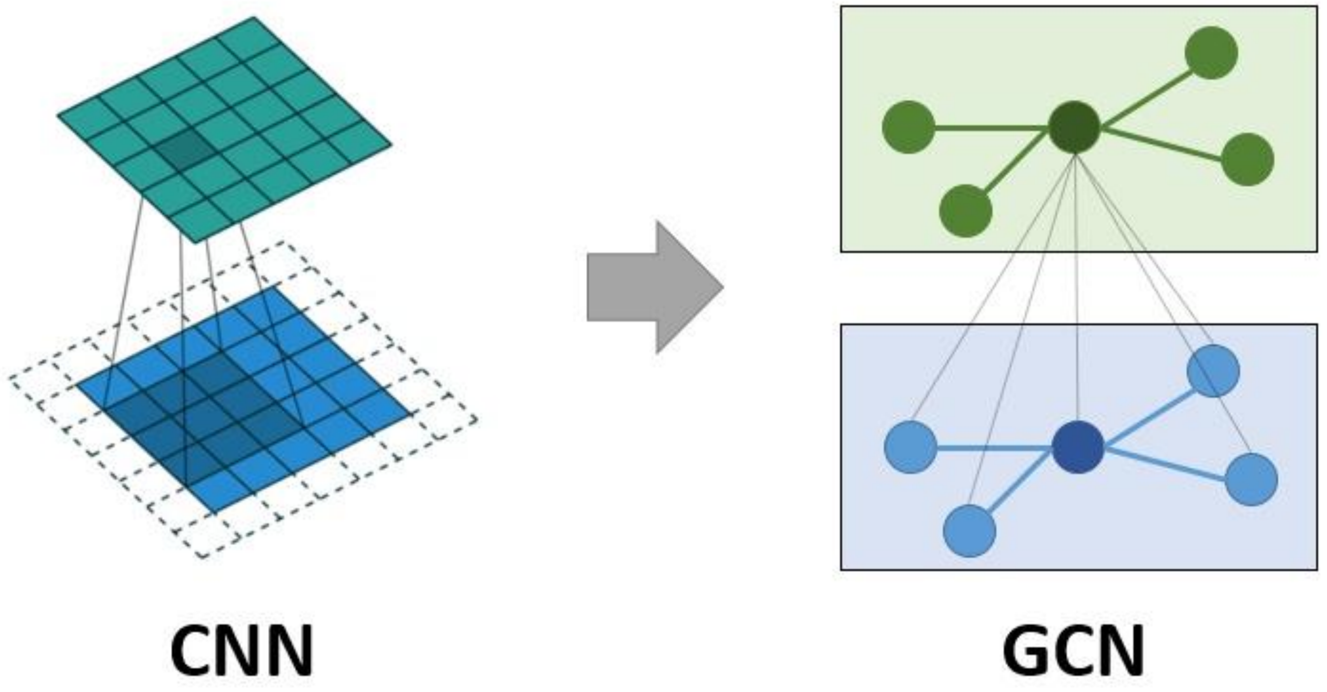
- Transform messages individually $\mathbf{W}_i\mathbf{h}_i$
- Add everything up $\sum_i \mathbf{W}_i\mathbf{h}_i$

$\mathbf{h}_i \in \mathbb{R}^F$ are (hidden layer) activations of a pixel/node

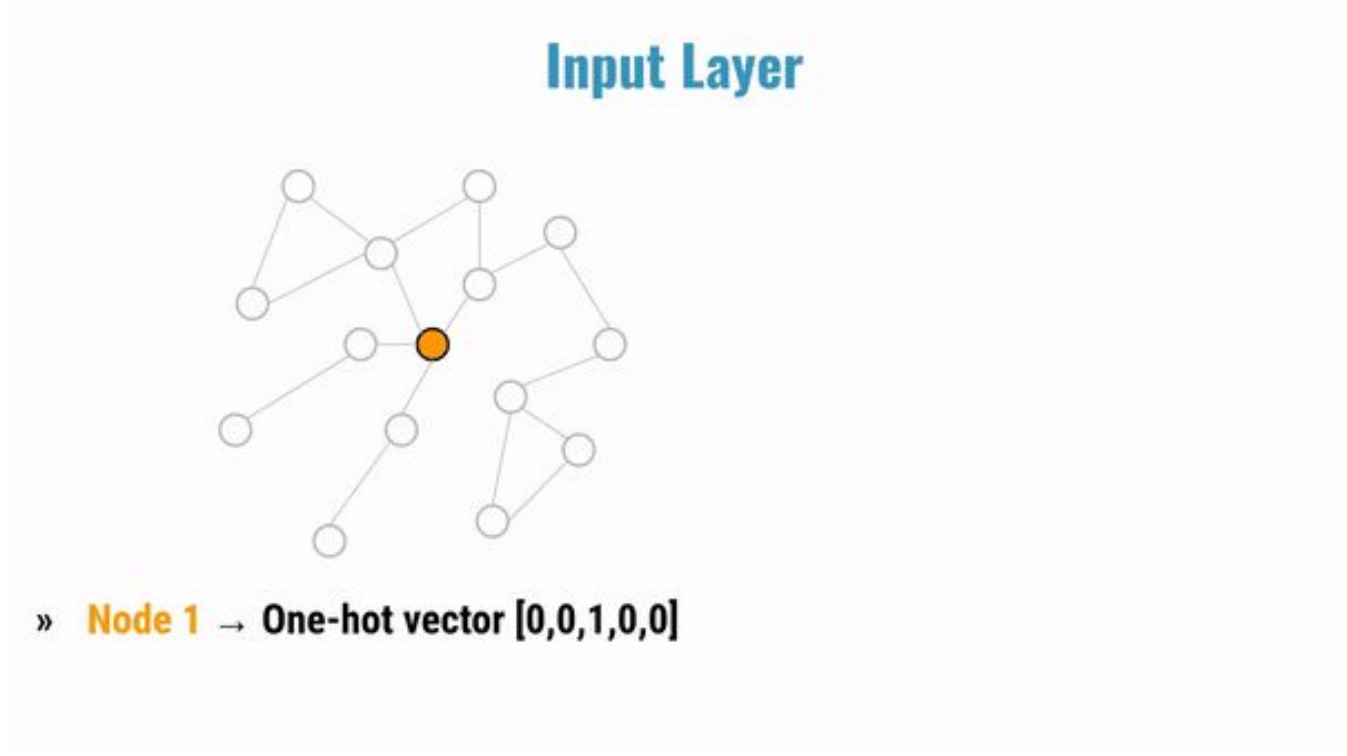
Full update:

$$\mathbf{h}_4^{(l+1)} = \sigma\left(\mathbf{W}_0^{(l)}\mathbf{h}_0^{(l)} + \mathbf{W}_1^{(l)}\mathbf{h}_1^{(l)} + \cdots + \mathbf{W}_8^{(l)}\mathbf{h}_8^{(l)}\right)$$

Comparing



Animations are good..



Formulation

- GNN formulation by [Kipf et al., ICLR 2016]

$$h_v = f\left(\frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} \underbrace{\boxed{W}}_{\substack{\text{Filter matrix}\\ \text{(Model-parameter)}}}\underbrace{\boxed{x_u}}_{\text{Initial Node Features}} + b\right), \quad \forall v \in \mathcal{V}.$$

Nodes == Words	$\rightarrow$ Word2vec Embeddings
Nodes == Authors	$\rightarrow$ 0/1 value indicating frequently used keywords
No features	$\rightarrow$ One-hot vector (length = #Nodes)

Formulation

- GNN formulation by [Kipf et al., ICLR 2016]

$$h_v = \boxed{f}\left(\underbrace{\boxed{\frac{1}{|\mathcal{N}(v)|}}}_{\text{Normalization}} \underbrace{\boxed{\sum_{u\in\mathcal{N}(v)} \underbrace{\boxed{W}}_{\text{Filter matrix (Model-parameter)}} \underbrace{\boxed{x_u}}_{\text{Initial Node Features}}}}_{\text{Neighborhood Aggregation}} + \underbrace{\boxed{b}}_{\text{Bias (Model-parameter)}}\right), \quad \forall v \in \mathcal{V}.$$

Formulation

- GNN formulation by [Kipf et al., ICLR 2016]

$$h_v = f\left(\frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} W x_u + b\right), \quad \forall v \in \mathcal{V}.$$

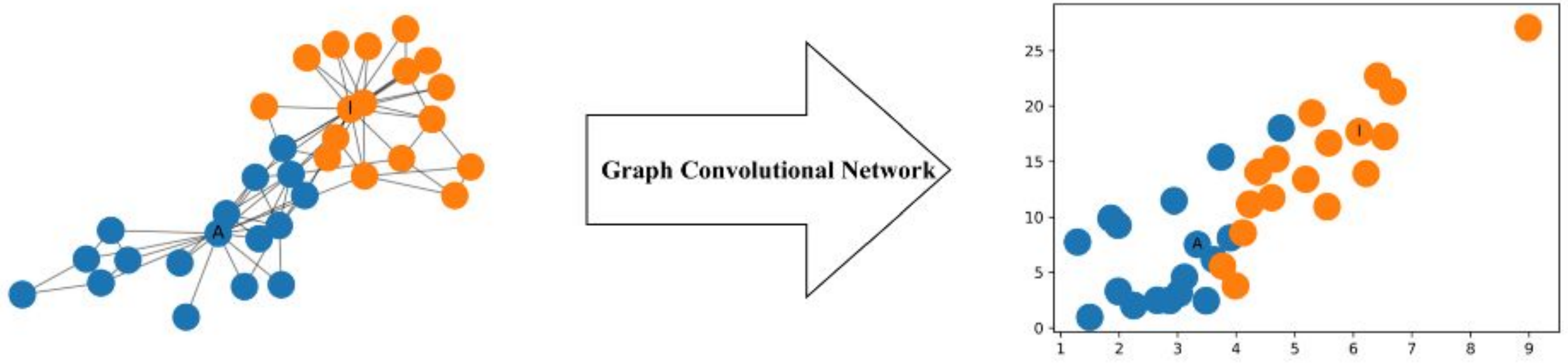
The above formulation is restricted to capturing just 1-hop of nodes

- Stacking K-GNN Layers for capturing K-hop nbd

$$h_v^{k+1} = f\left(\frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} W^k \boxed{h_u^k} + b^k\right), \quad \forall v \in \mathcal{V}.$$

Hello World in Deep Learning on Graph

Let's Understand by practical examples :



Hello World in Deep Learning on Graph

Given a graph G = (V, E), a GCN takes as input

- An input feature matrix $N \times F^0$ feature matrix, $\boldsymbol{X}$, where N is the number of nodes and F^0 is the number of input features for each node
- $N \times N$ matrix representation of the graph structure such as the adjacency matrix $\boldsymbol{A}$ of G.

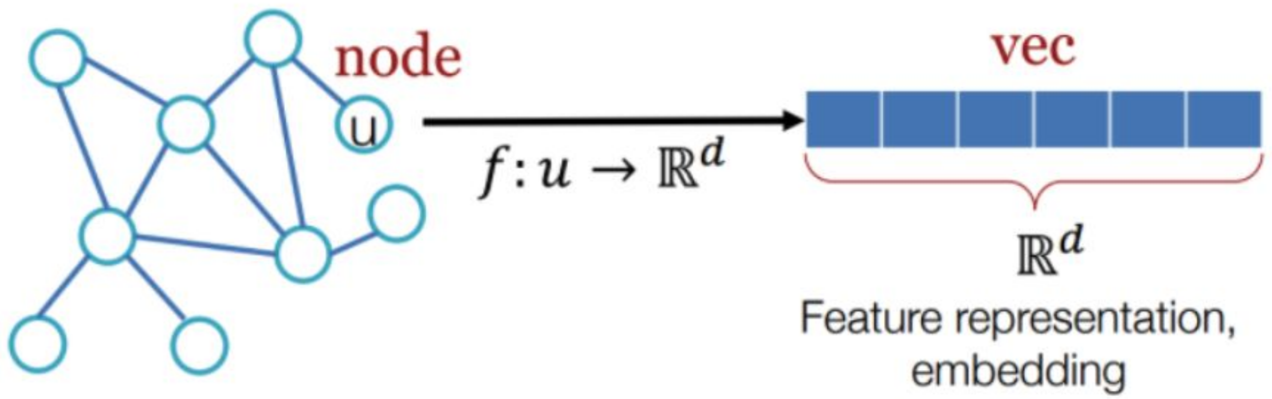
Hello World in Deep Learning on Graph

A Simple Propagation Rule

$$f(H^i, A) = \sigma(AH^iW')$$

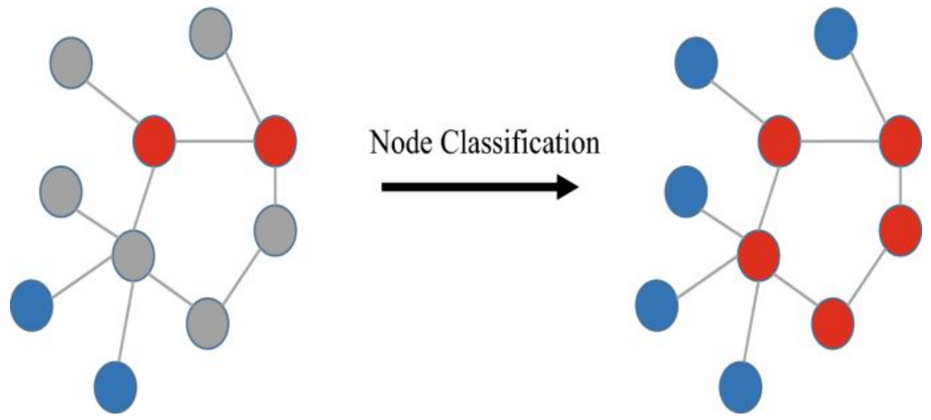
Hello World in Deep Learning on Graph

Every Node is a vector of *d* dimension

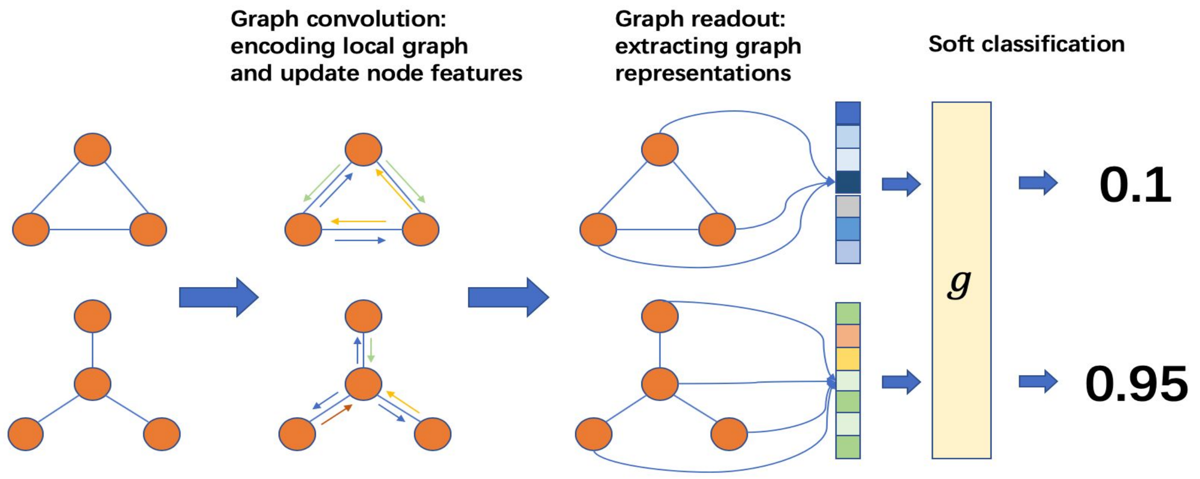

node
u
$f : u \rightarrow \mathbb{R}^d$
vec
$\mathbb{R}^d$
Feature representation, embedding

Application of GNN

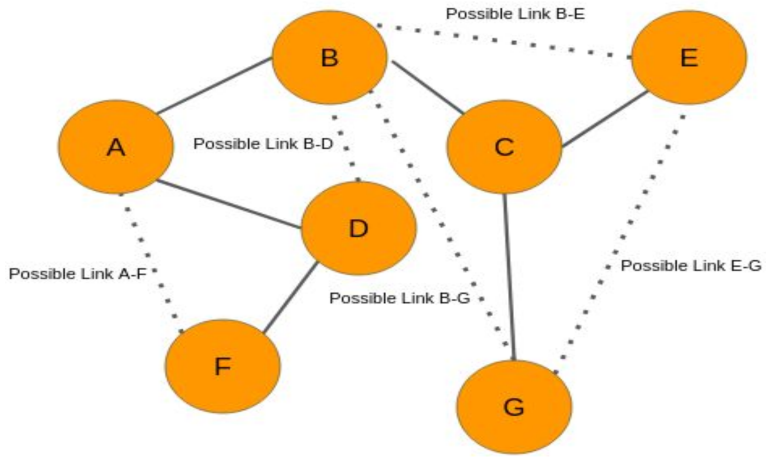
Node Classification



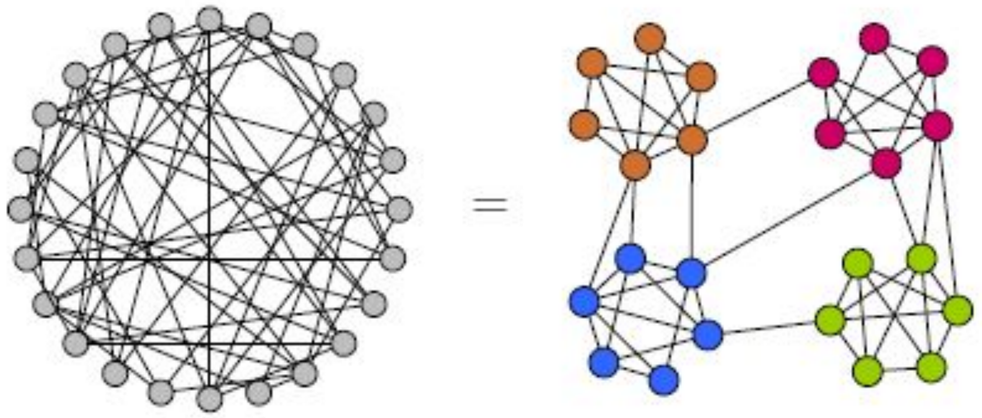
Graph Classification


Graph convolution: encoding local graph and update node features
Graph readout: extracting graph representations
Soft classification
g
0.1
0.95

Link Prediction



Graph Clustering



Example

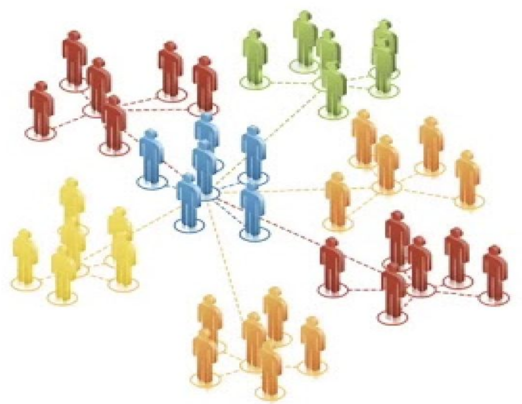
Jupyter Notebook explanation : Notebook 1

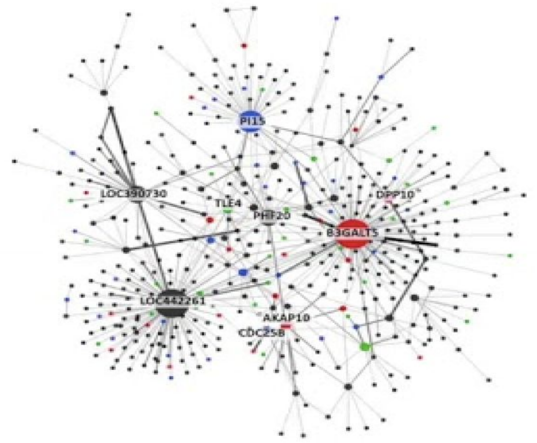
https://tinyurl.com/gcn-formulation

Key Points from Notebook

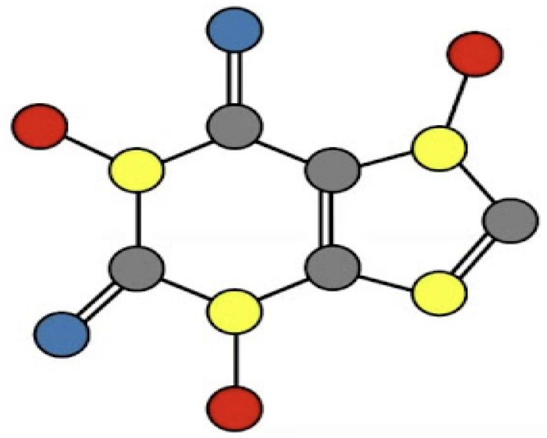
- In Graph based Models, using the Adjacency Matrix(A) enables the model to learn the features of neighboring nodes.
- Thomas Kipf and Max Welling (2017) Used the Renormalization trick to normalize the features in Fast Approximate Spectral-based Graph Convolutional Networks.
- GCNs can learn features representation to cluster even before training.

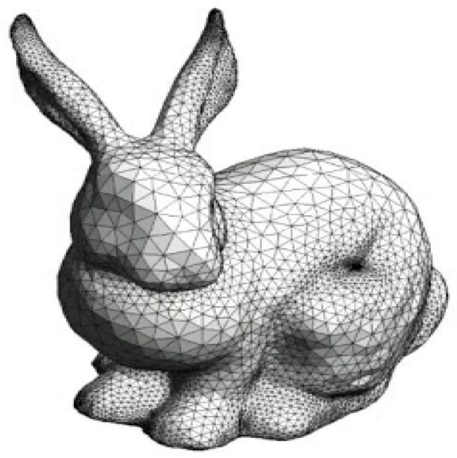
Applications of Geometric Deep Learning

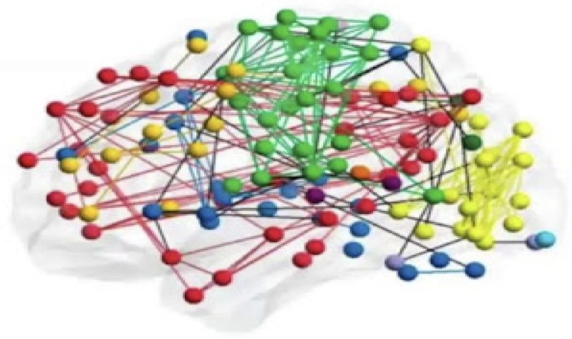

Social networks



Interaction networks

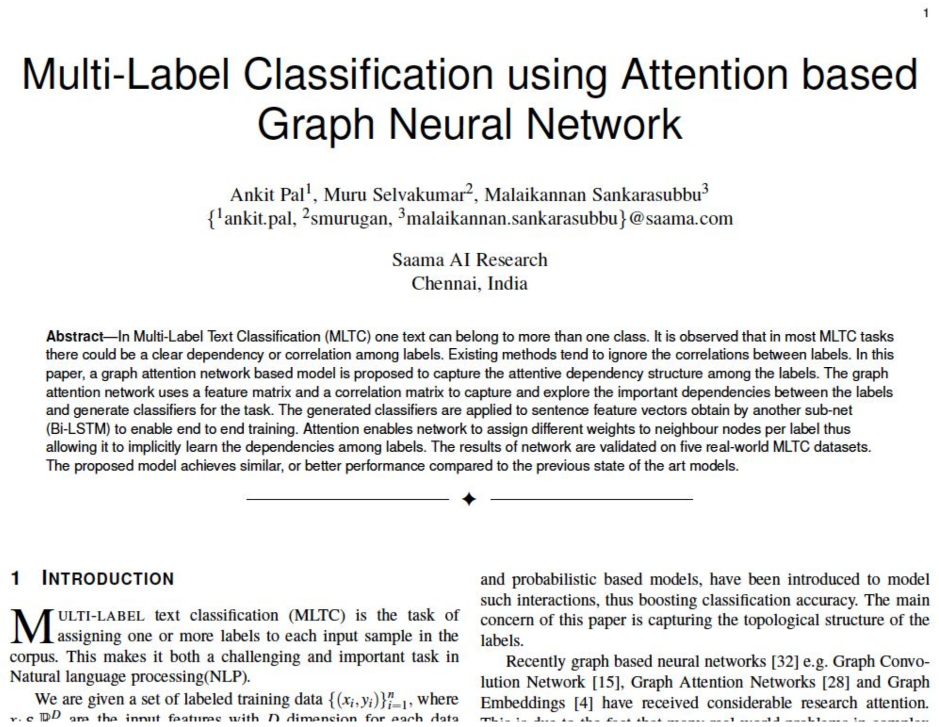

Molecules


Meshes


Functional networks

Applications

MAGNET : Multi-Label Text Classification using Attention-based Graph Neural Network



Multi-Label Classification using Attention based Graph Neural Network

Ankit Pal[1], Muru Selvakumar[2], Malaikannan Sankarasubbu[3]
{[1]ankit.pal, [2]smurugan, [3]malaikannan.sankarasubbu}@saama.com

Saama AI Research
Chennai, India

Abstract—In Multi-Label Text Classification (MLTC) one text can belong to more than one class. It is observed that in most MLTC tasks there could be a clear dependency or correlation among labels. Existing methods tend to ignore the correlations between labels. In this paper, a graph attention network based model is proposed to capture the attentive dependency structure among the labels. The graph attention network uses a feature matrix and a correlation matrix to capture and explore the important dependencies between the labels and generate classifiers for the task. The generated classifiers are applied to sentence feature vectors obtain by another sub-net (Bi-LSTM) to enable end to end training. Attention enables network to assign different weights to neighbour nodes per label thus allowing it to implicitly learn the dependencies among labels. The results of network are validated on five real-world MLTC datasets. The proposed model achieves similar, or better performance compared to the previous state of the art models.

✦

1 Introduction

MULTI-LABEL text classification (MLTC) is the task of assigning one or more labels to each input sample in the corpus. This makes it both a challenging and important task in Natural language processing(NLP).

We are given a set of labeled training data $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^D$ are the input features with D dimension for each data

and probabilistic based models, have been introduced to model such interactions, thus boosting classification accuracy. The main concern of this paper is capturing the topological structure of the labels.

Recently graph based neural networks [32] e.g. Graph Convolution Network [15], Graph Attention Networks [28] and Graph Embeddings [4] have received considerable research attention. This is due to the fact that many real world problems in complex

https://www.groundai.com/project/multi-label-text-classification-using-attention-based-graph-neural-network/1

Credits

1. Extra Credits: The History of Non-Euclidean Geometry
2. Michael Bronstein: Geometric Deep Learning: Going beyond Euclidean data, Geometric Deep Learning Grids, Groups, Graphs, Geodesics, and Gauges
3. Ankit Kumar Pal: MAGNET