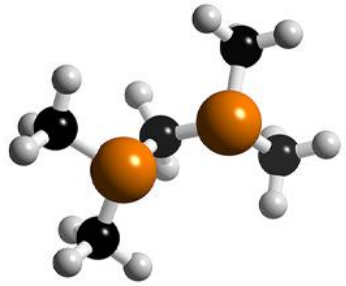
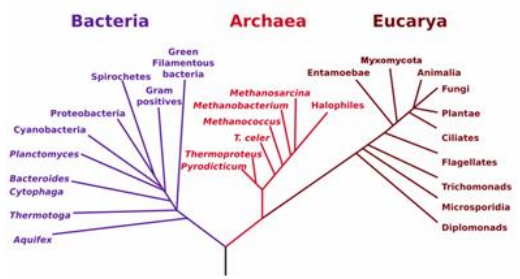
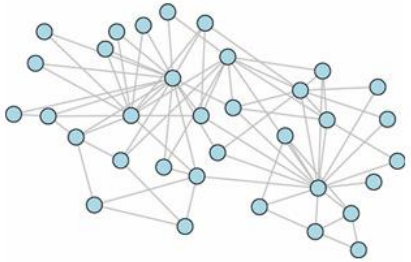
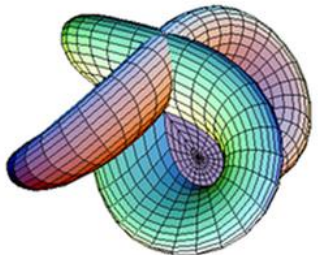
Geometric deep learning with graph neural networks









Deep Learning ❤ Euclidean data

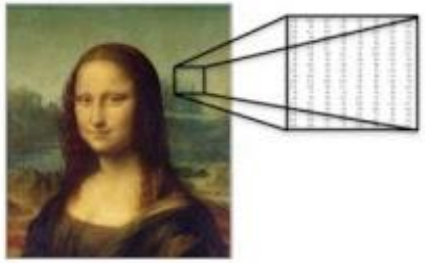
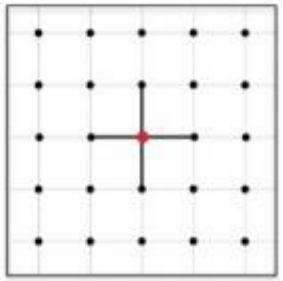
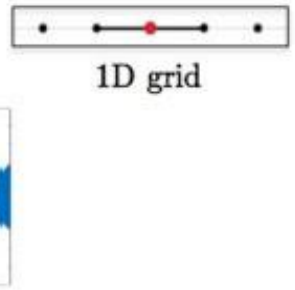


Image.



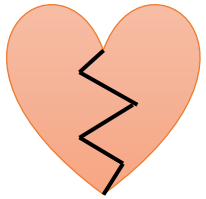
2D grids

Sentence.



sound:

1D grid

Deep Learning  Non-Euclidean data

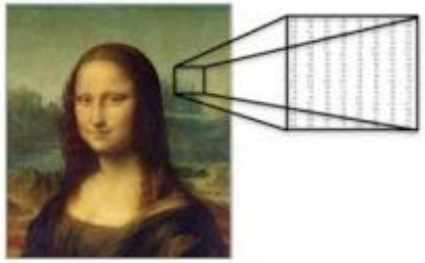
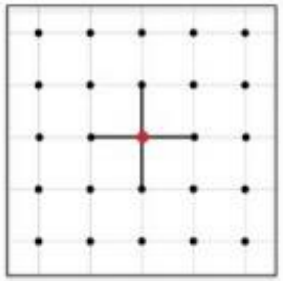
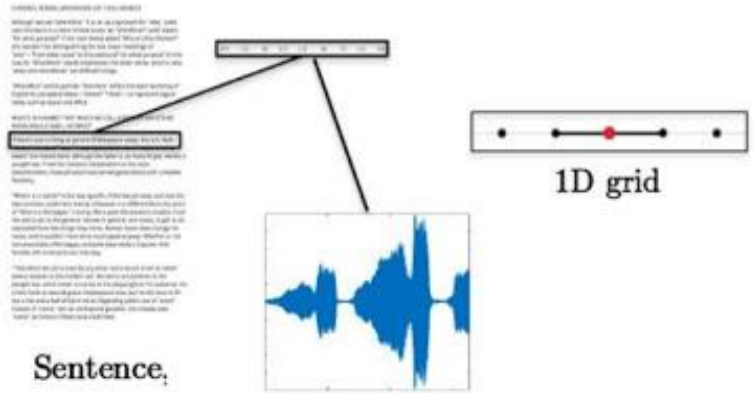


Image.



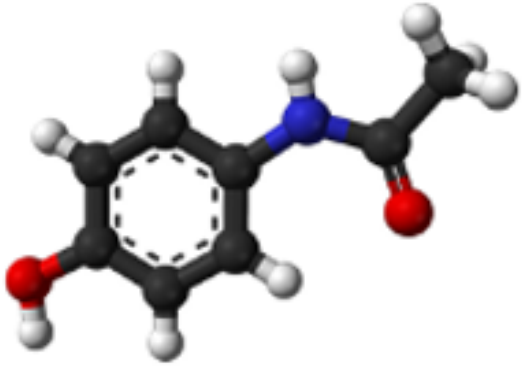
2D grids



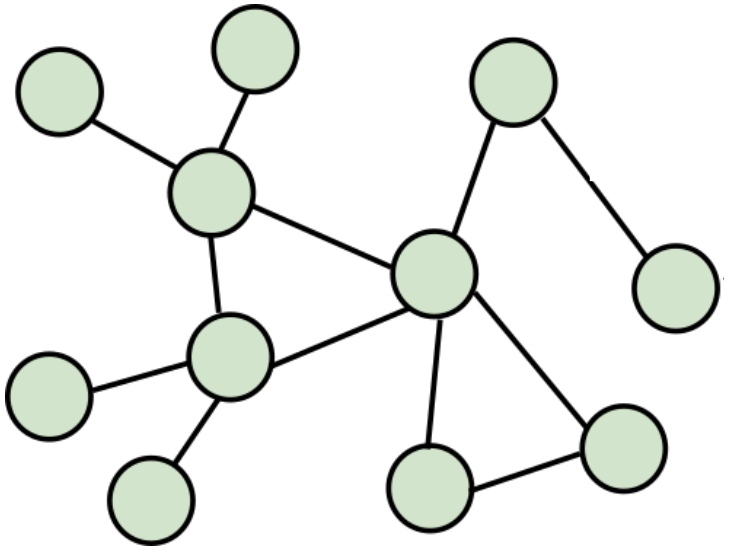
Sentence.

sound:

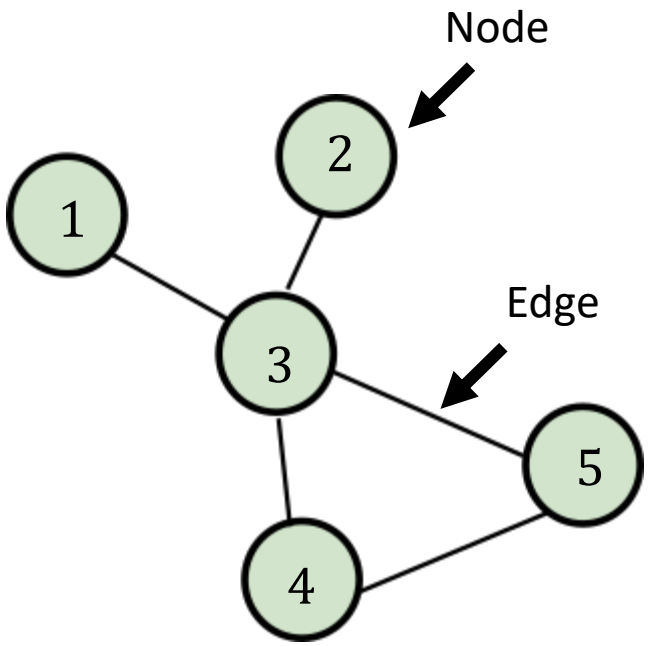
1D grid



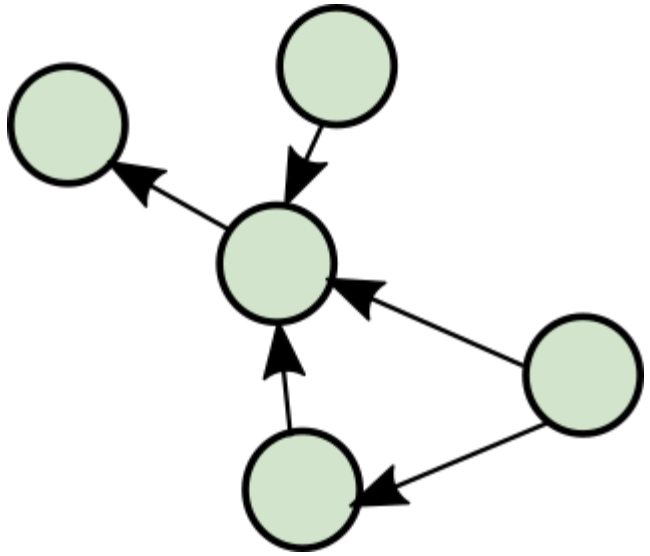
Graphs to the rescue



Basic concepts



Undirected

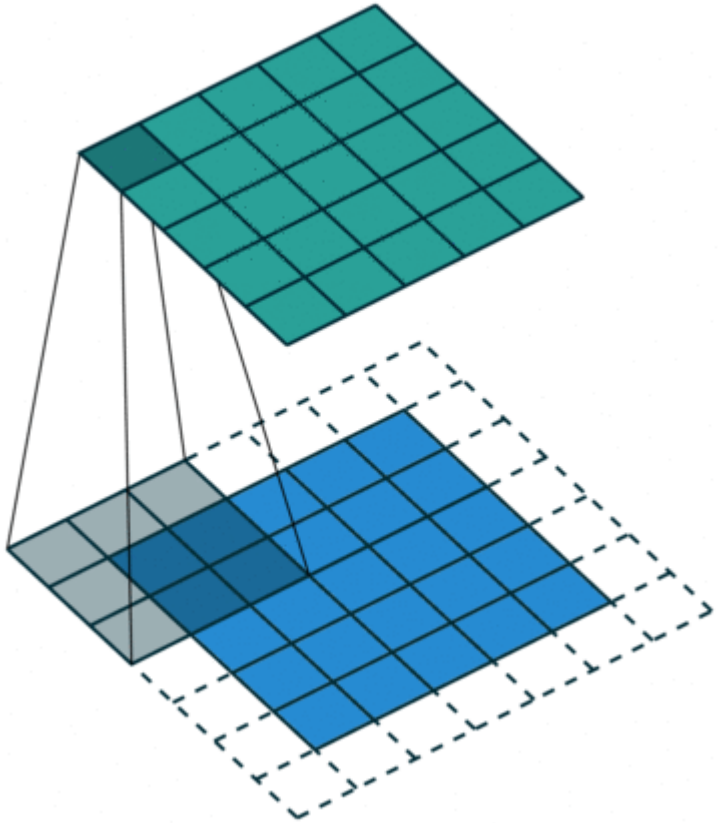


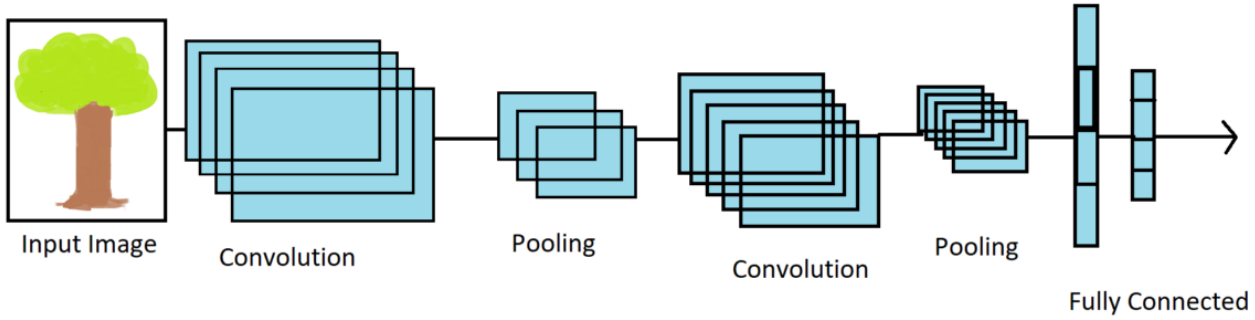
Directed

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Adjacency matrix

Convolutional Neural Network (CNN)





Vincent Dumoulin, Francesco Visin - A guide to convolution arithmetic for deep learning

Image = Graph

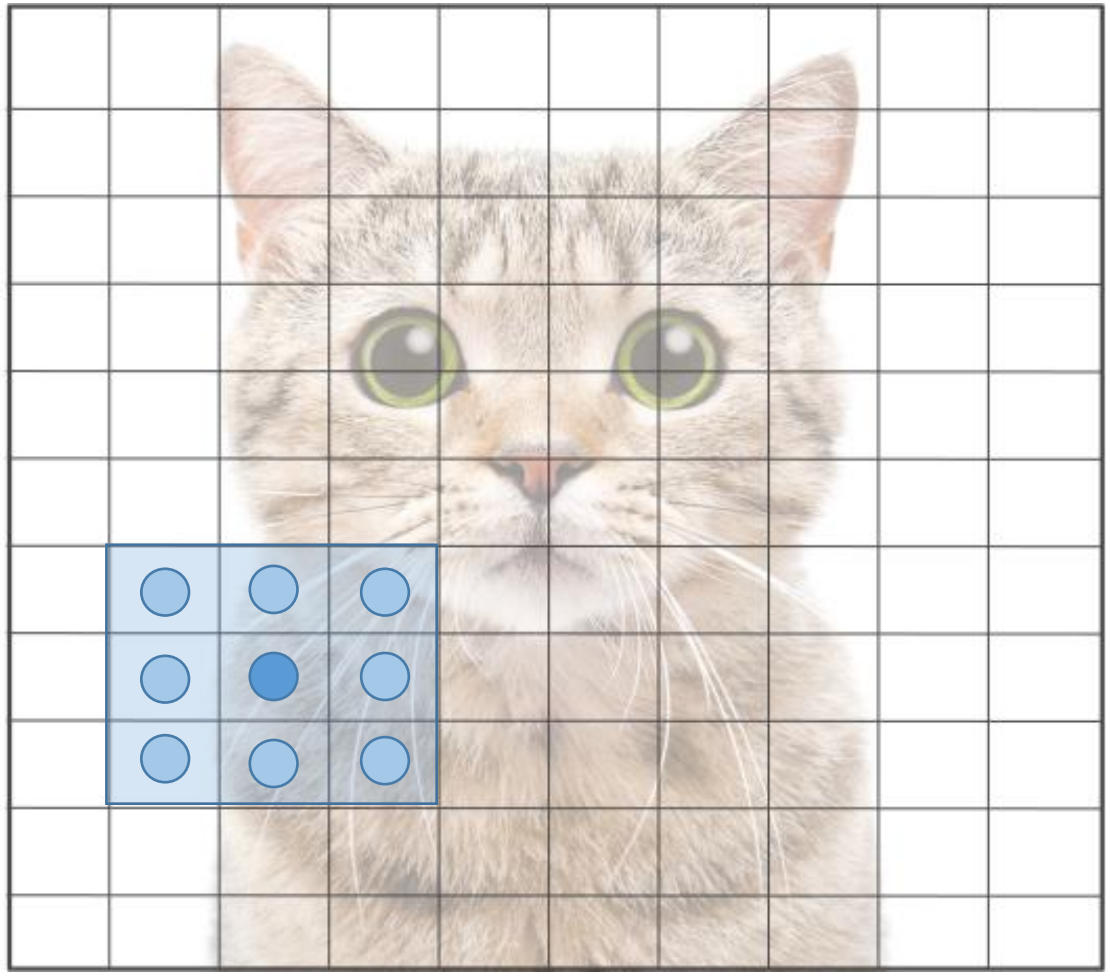
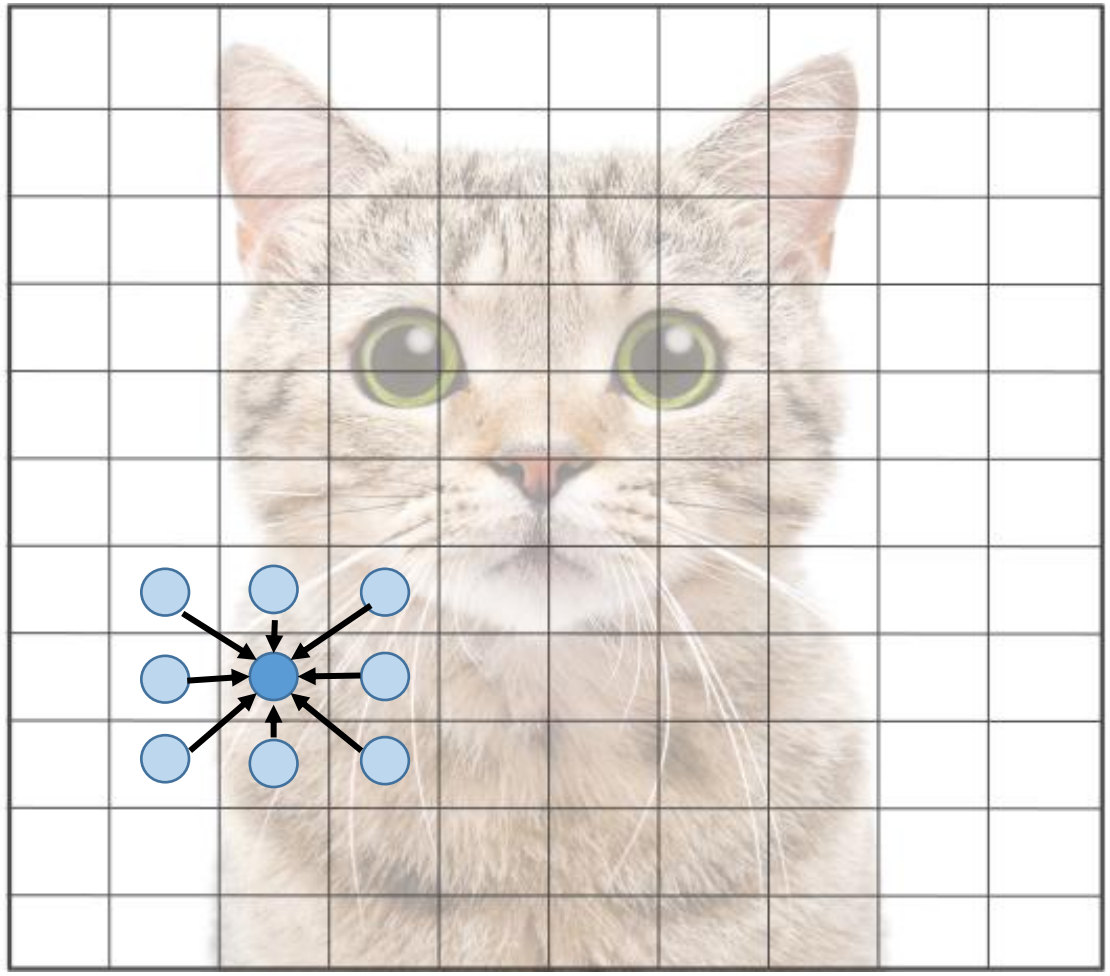
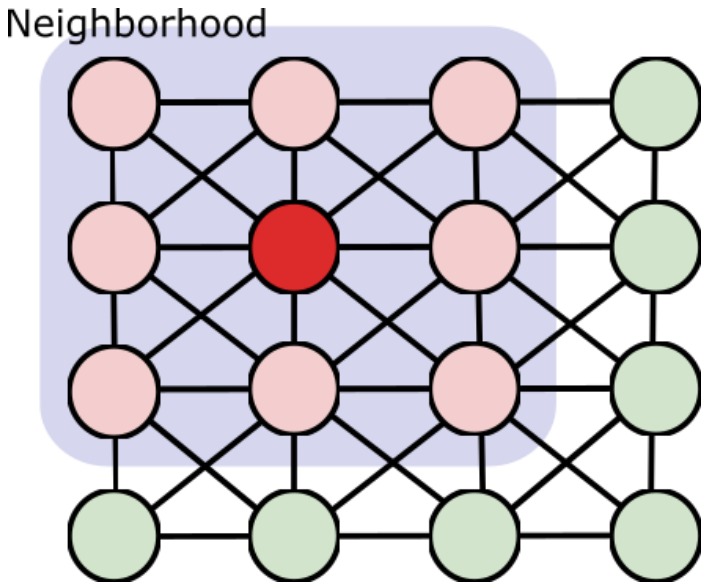
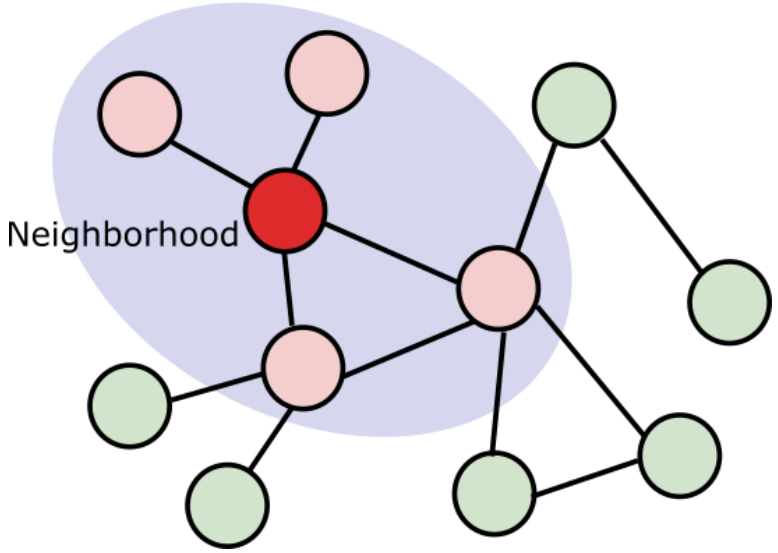


Image = Graph

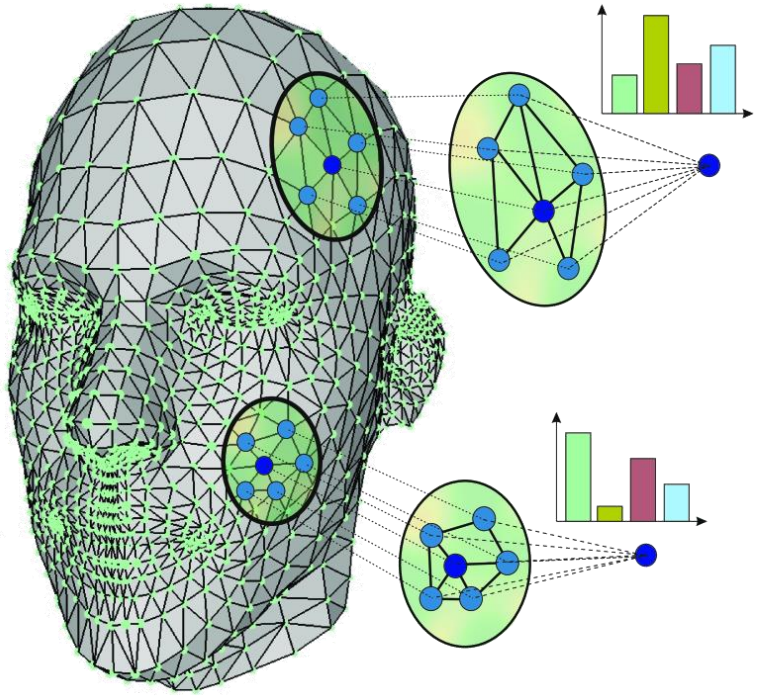


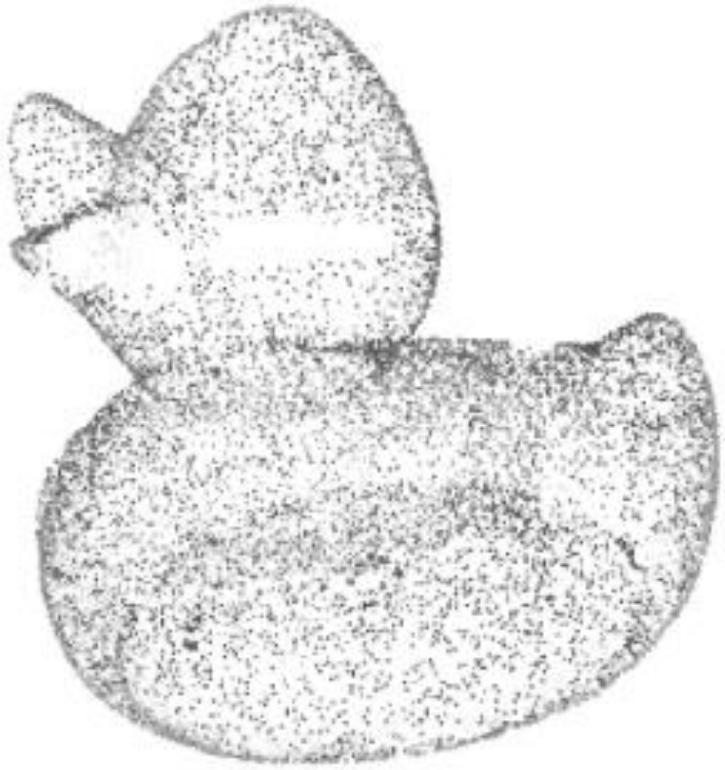
Convolutions



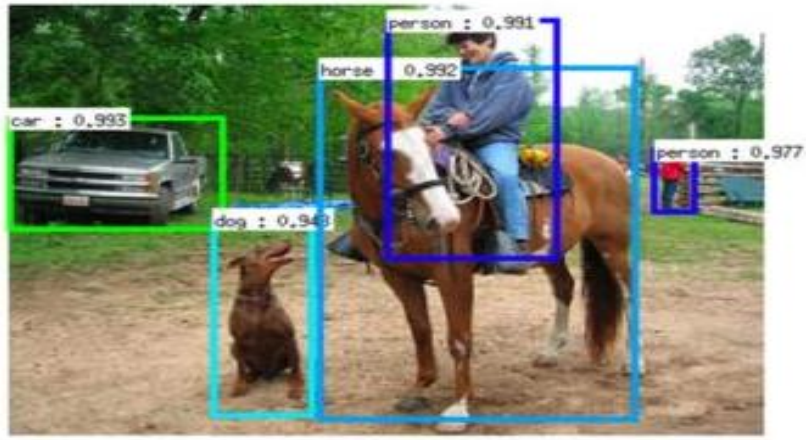


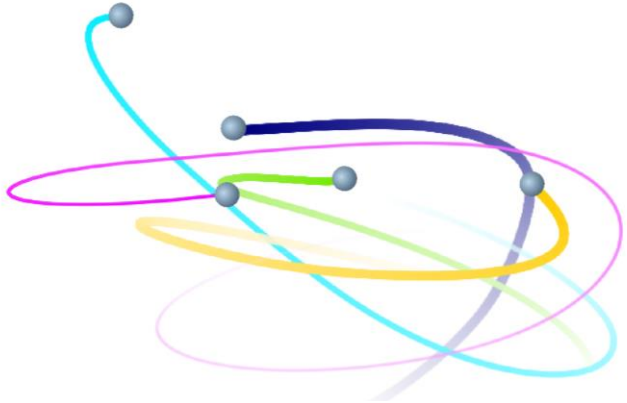
Curved surfaces

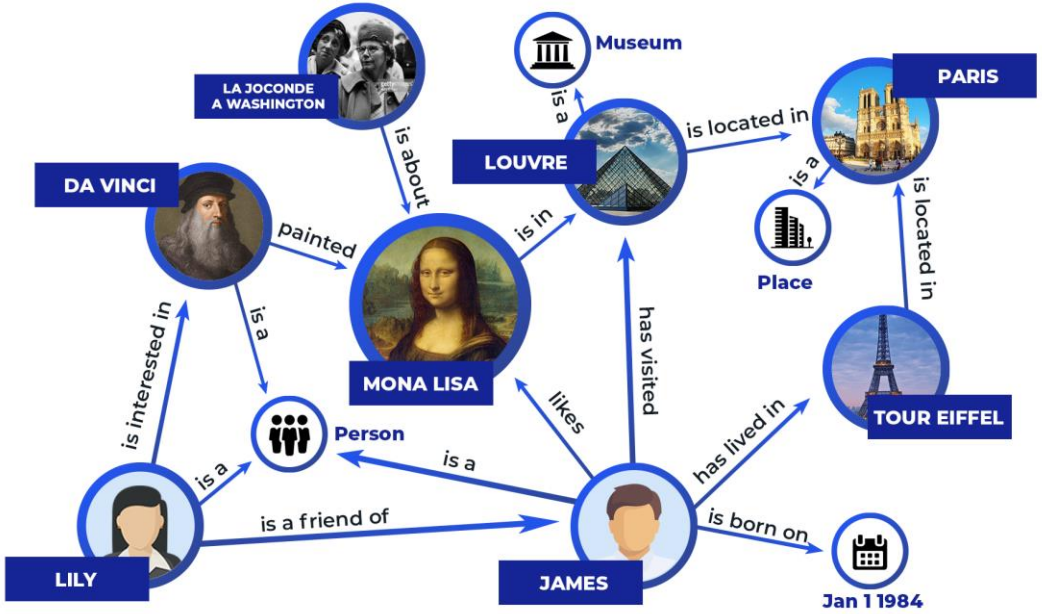




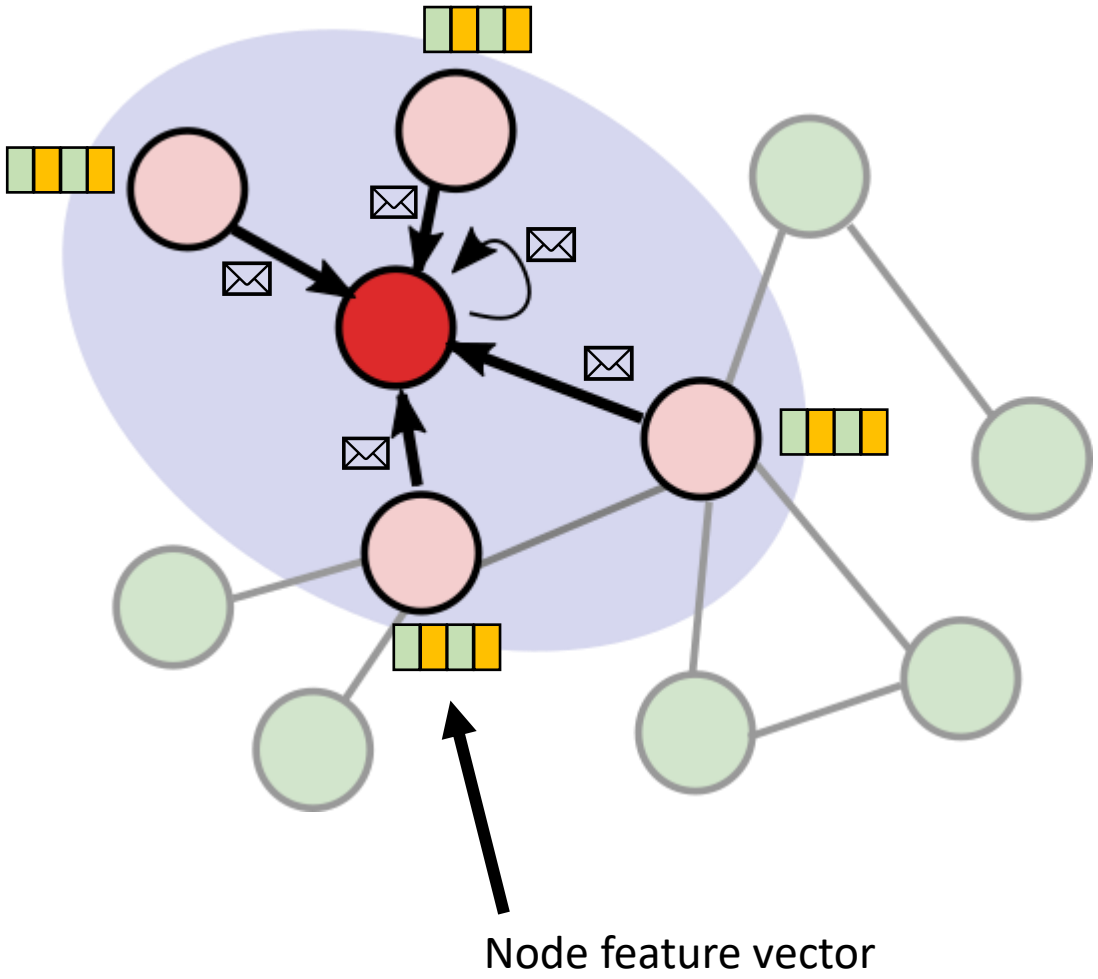
Objects with relations



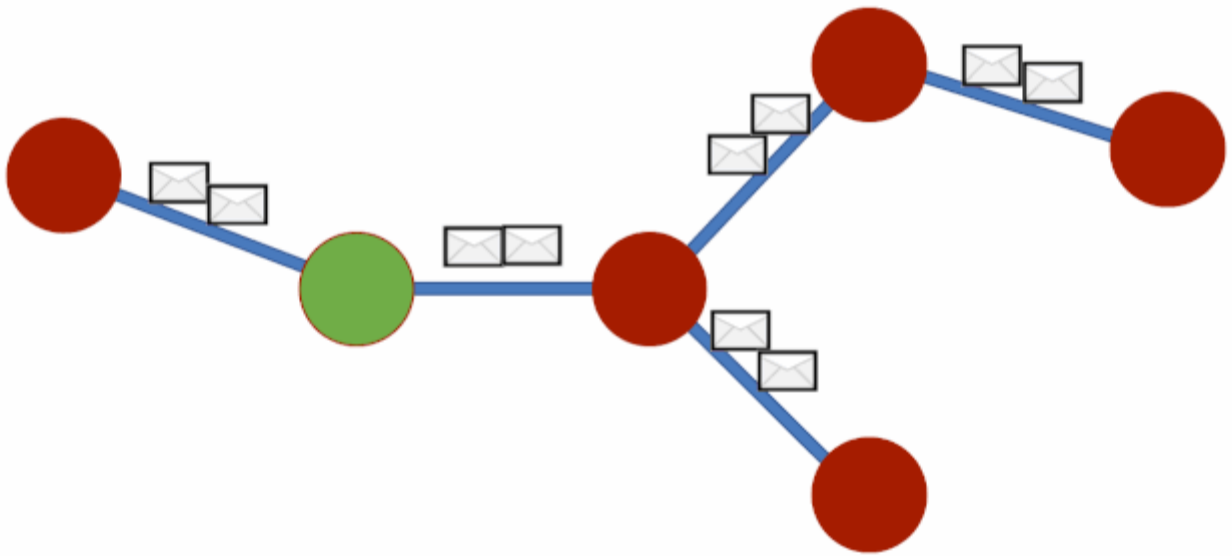




Convolution = Message passing



Message passing layer



https://www.outsystems.com/blog/posts/graph-neural-networks/

Message passing

Message from node $j \rightarrow i$:

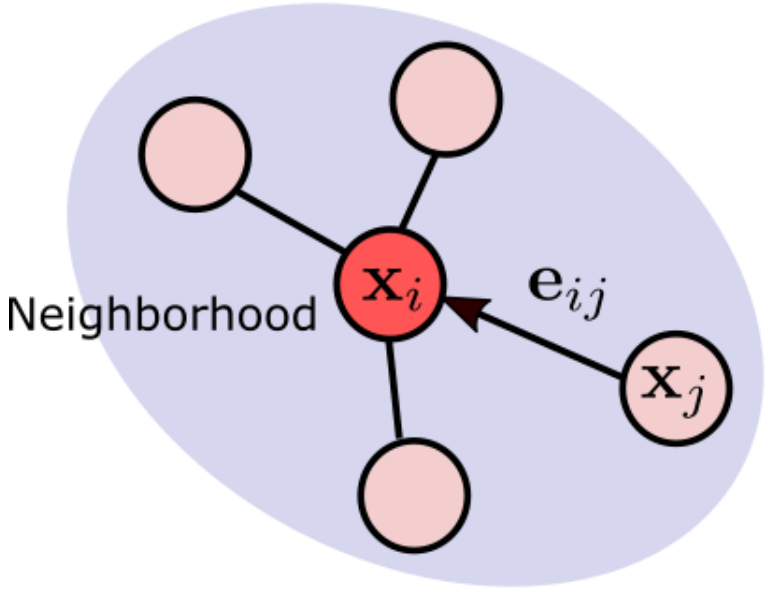
$$\boldsymbol{m}_j = m(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{e}_{ij})$$

Aggregate messages from all neighbors:

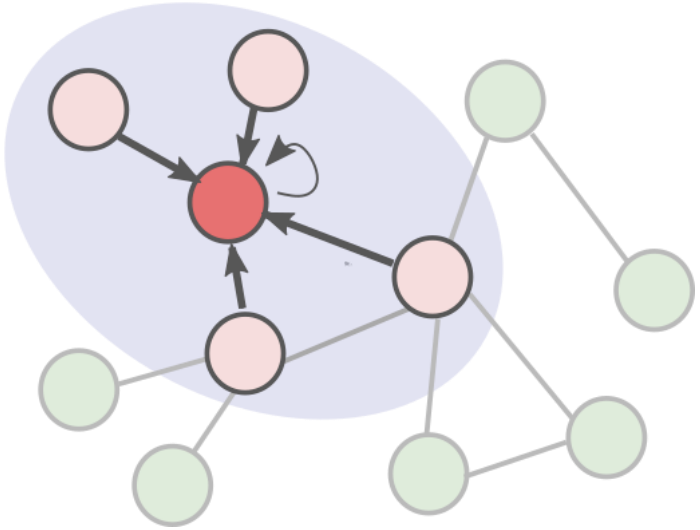
$$\boldsymbol{a} = a(\boldsymbol{m}_j | \ j \ \ neighborhood\ of\ i\ )$$

Update node i:

$$\boldsymbol{x}_i' = u(\boldsymbol{x}_i, \boldsymbol{a})$$

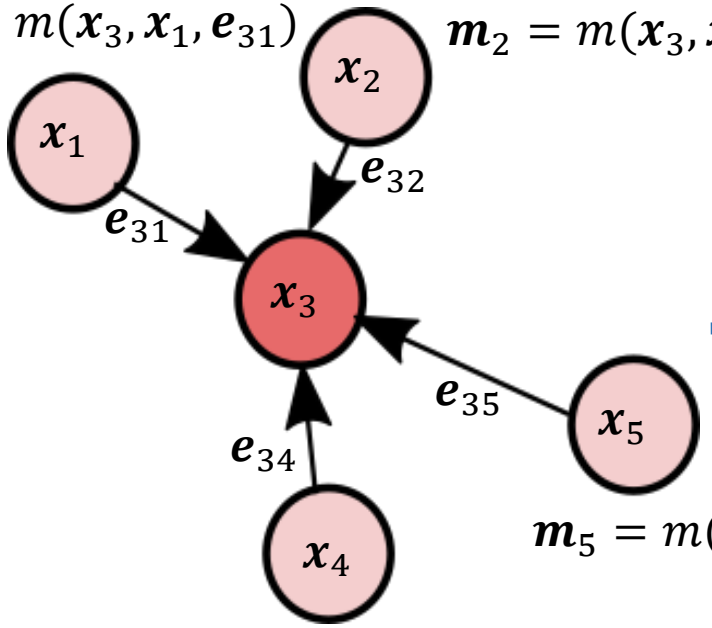


Message passing



Message

$\boldsymbol{m_1 = m(x_3, x_1, e_{31})}$ $\boldsymbol{m_2 = m(x_3, x_2, e_{32})}$



$\boldsymbol{m_5 = m(x_3, x_5, e_{35})}$

$\boldsymbol{m_4 = m(x_3, x_4, e_{34})}$

Aggregate (mean)

$$\boldsymbol{a} = \frac{\boldsymbol{m_1} + \boldsymbol{m_2} + \boldsymbol{m_4} + \boldsymbol{m_5}}{4}$$

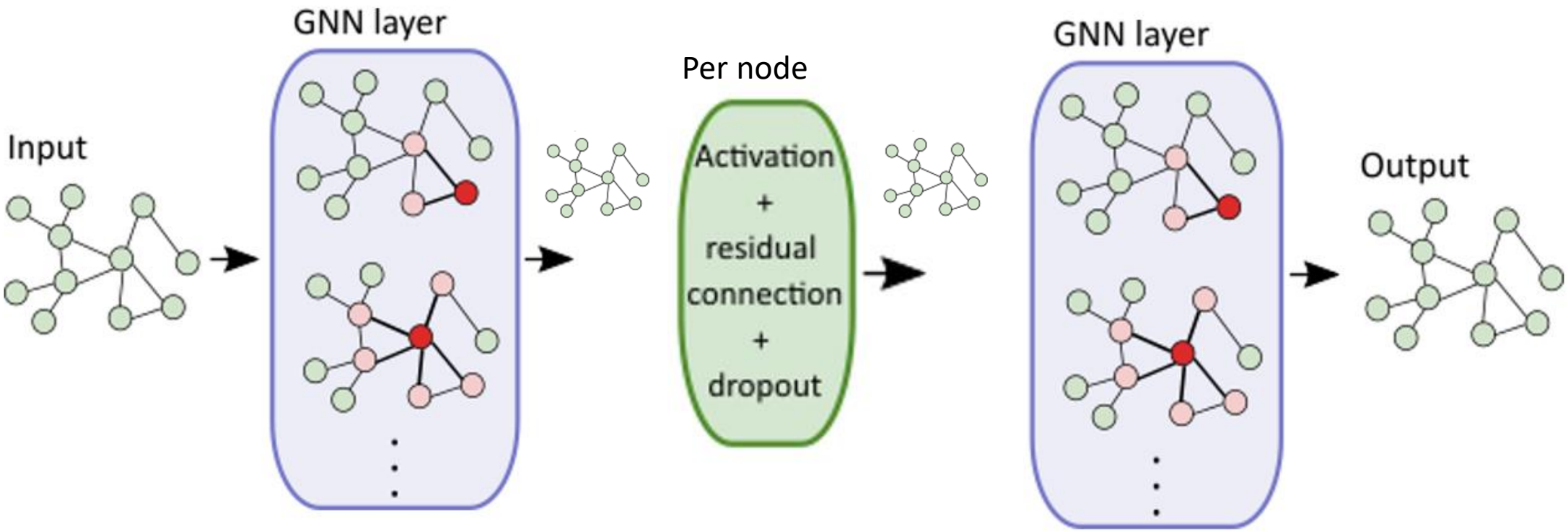
$\boldsymbol{a}$

Update

$u(\boldsymbol{x_3}, \boldsymbol{a})$

$\boldsymbol{x_3'}$

Two layer GNN



Graph Convolutional Network

Thomas N Kipf and Max Welling 2017

Message from node $j \rightarrow i$:

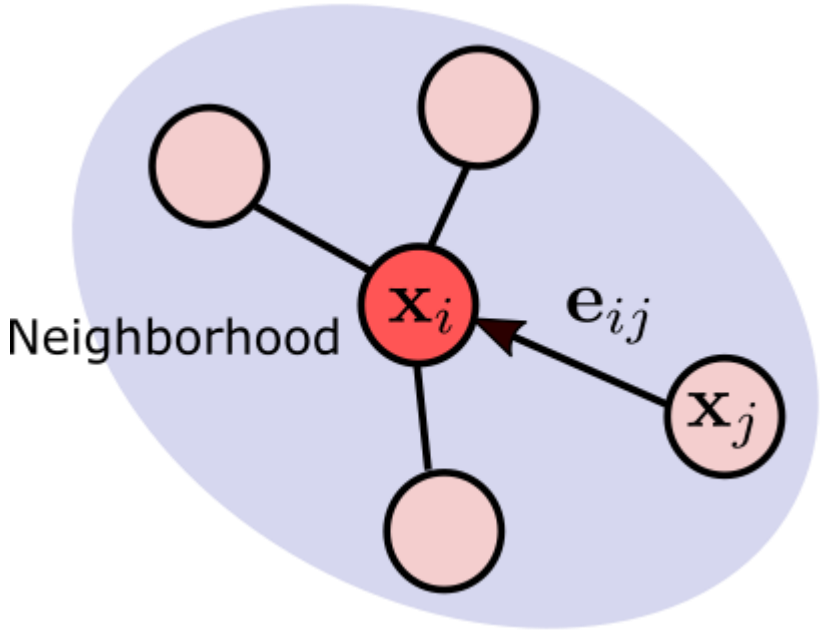
$$\boldsymbol{m}_j^k = \frac{\boldsymbol{e}_{ij}}{\sqrt{d_i d_j}} \boldsymbol{x}_j^k, \qquad d_i = 1 + \sum_{j \in N(i)} \boldsymbol{e}_{ij}$$

Aggregate messsages from all neighbors:

$$\boldsymbol{a}^k = \sum_{\boldsymbol{j}} \boldsymbol{m}_j^k$$

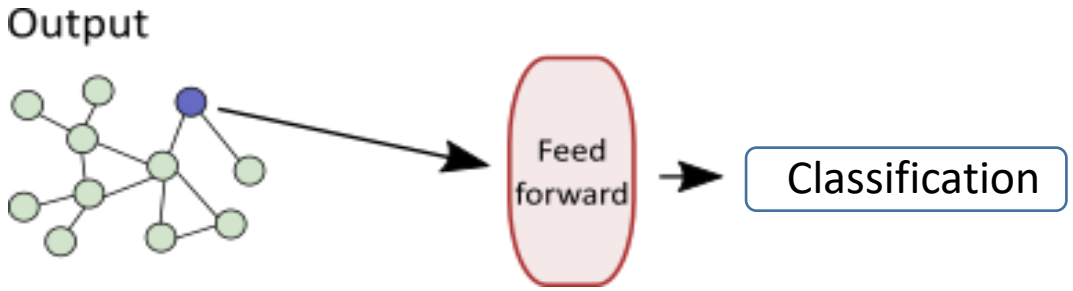
Update node i:

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{A}\boldsymbol{a}^k$$

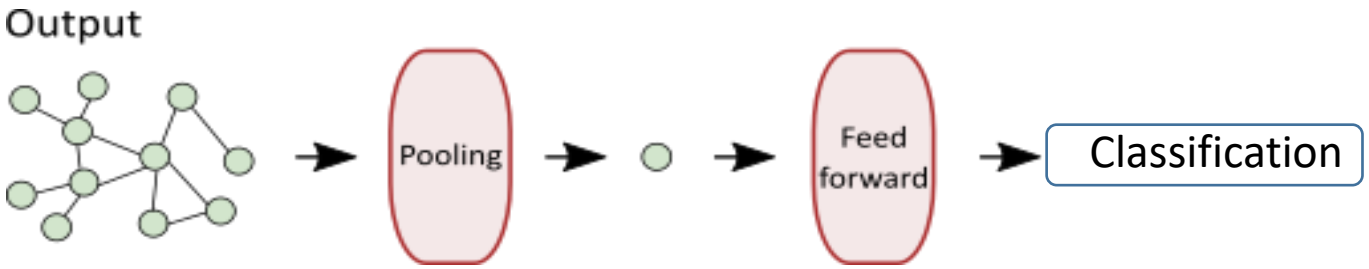


Supervised learning

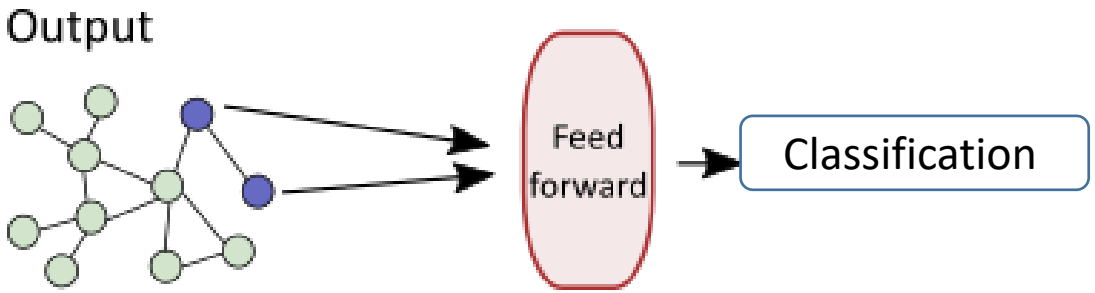
Node classification:



Graph classification:

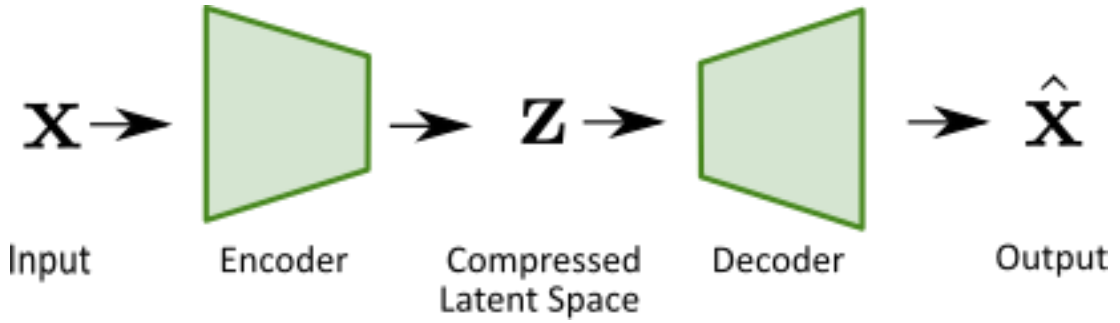


Link prediction:



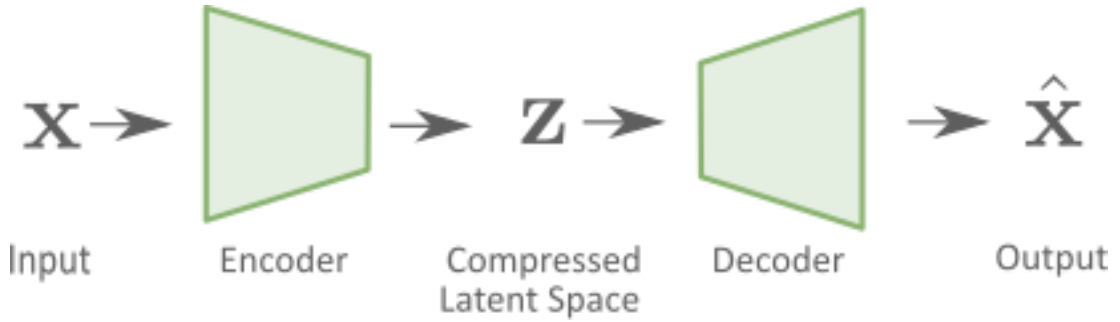
Graph Auto Encoders

Traditional Auto Encoder:

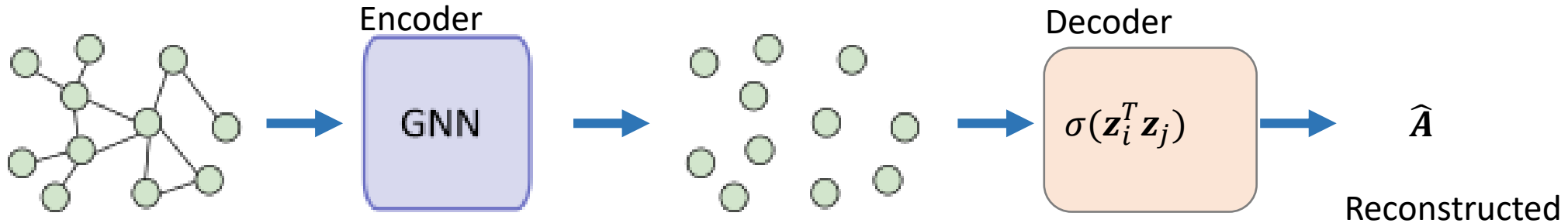


Graph Auto Encoders

Traditional Auto Encoder:



Graph Auto Encoder:



Node features: $X \in \mathbb{R}^{N \times D}$

Adjacency matrix: $A \in \{0,1\}^{N \times N}$

Graph embedding $Z \in \mathbb{R}^{N \times F}$

Reconstructed adjacency matrix $\hat{A} \in \{0,1\}^{N \times N}$

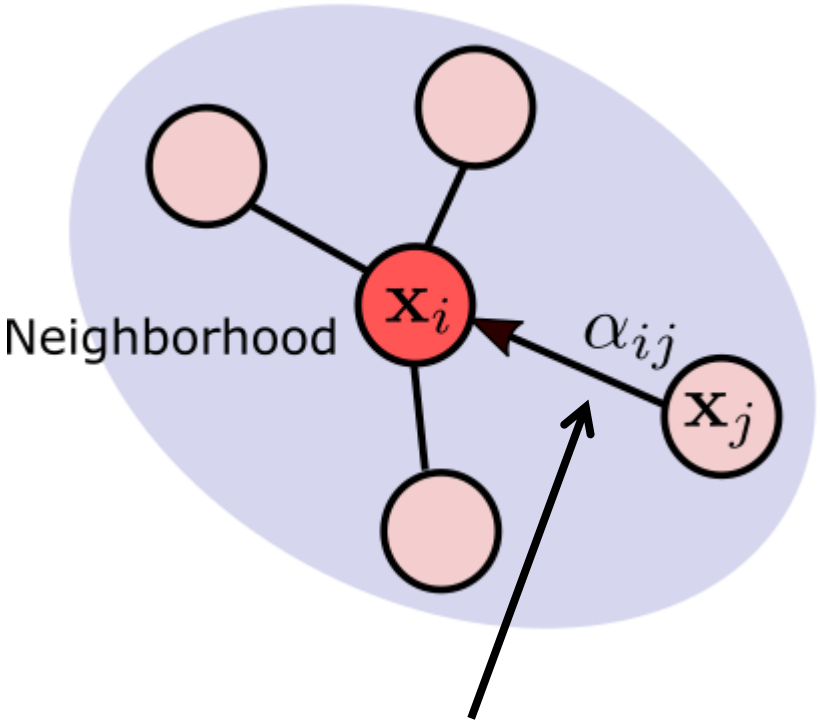
Variational Graph Auto-Encoders by Thomas N Kipf and Max Welling

Attention in GNN

Graph Attention Network (Vaswani et al NIPS 2017)

Message $\boldsymbol{m}_j = m(\boldsymbol{x}_i, \boldsymbol{x}_j)$

$\alpha_{ij} = f(\boldsymbol{x}_i, \boldsymbol{x}_j)$ Attention score



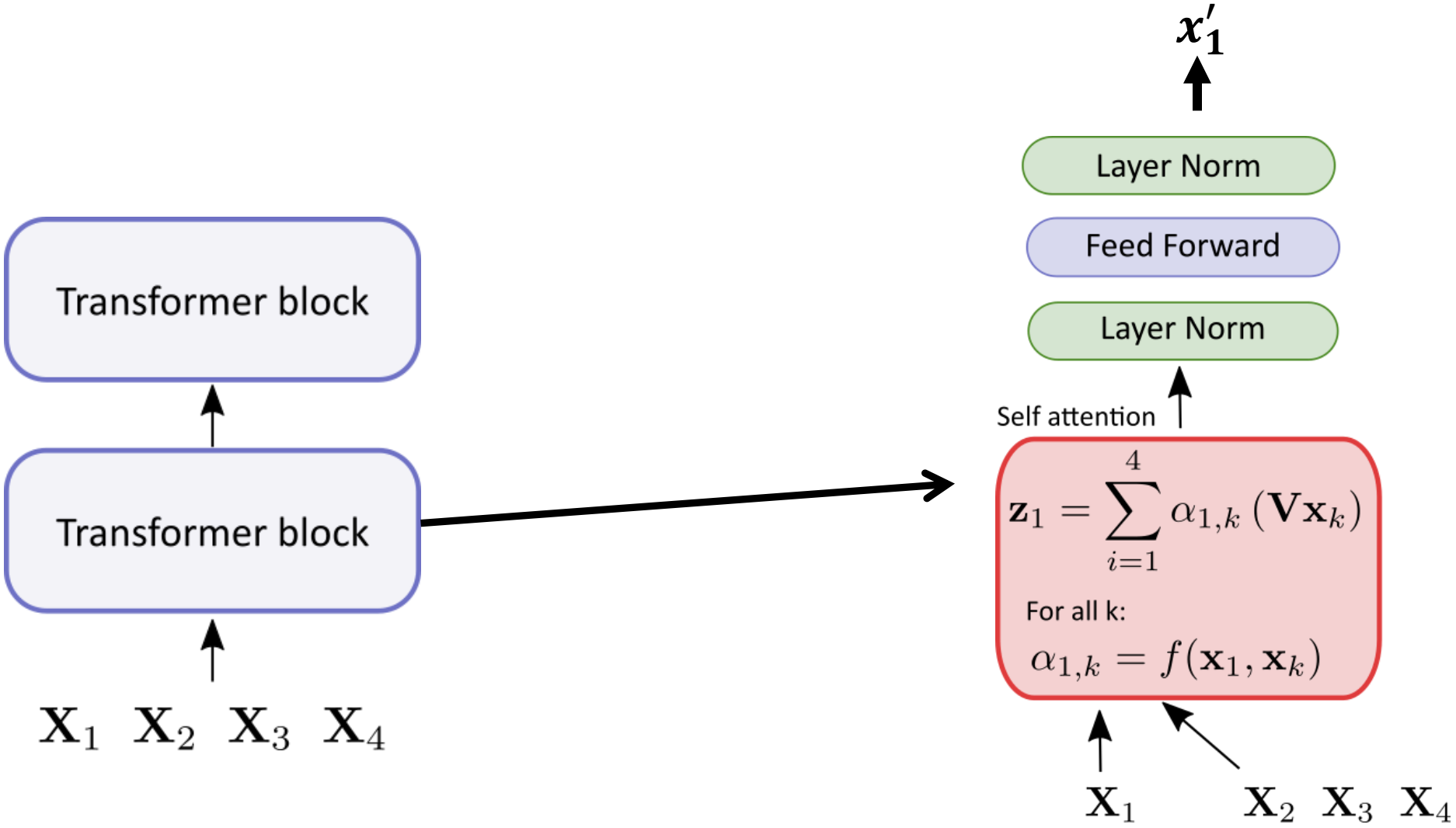
Aggregation function

$$\boldsymbol{a} = \sum_{j \in N(i)} \alpha_{ij} \boldsymbol{m}_j$$

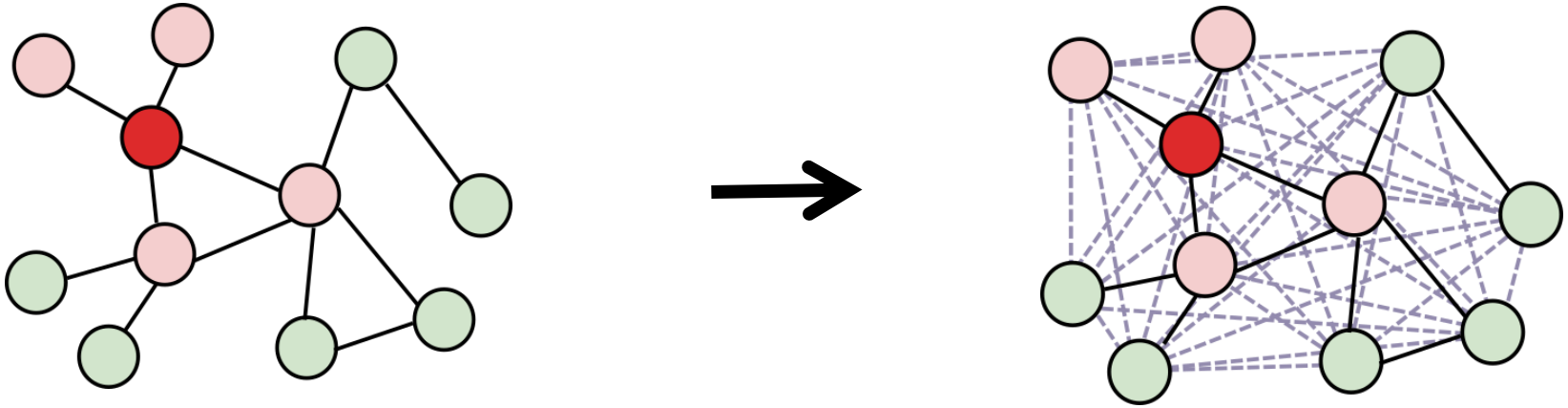
$neighborhood\ of\ i$

$\alpha_{ij} = f(\boldsymbol{x}_i, \boldsymbol{x}_j)$

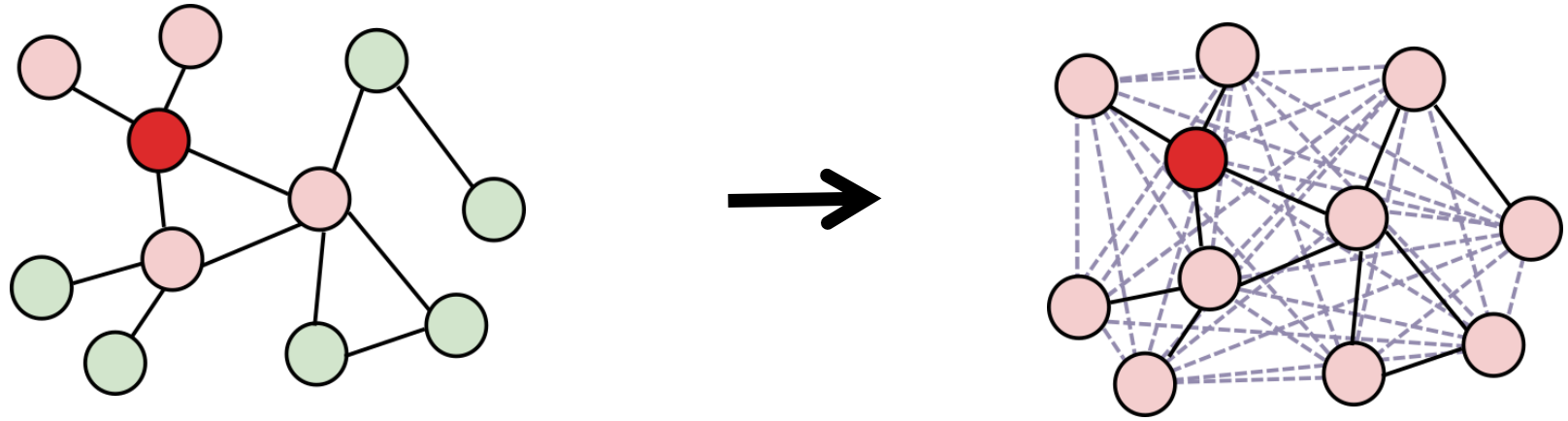
GNN vs Transformers



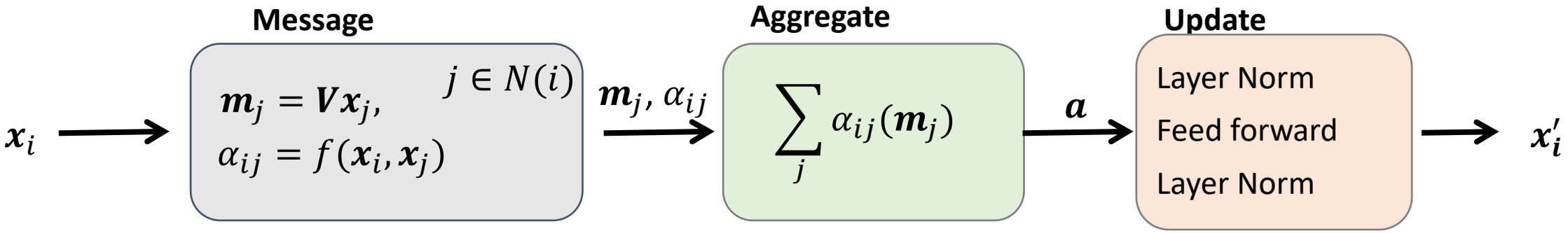
Graph neural network vs Transformers



Graph neural network vs Transformers

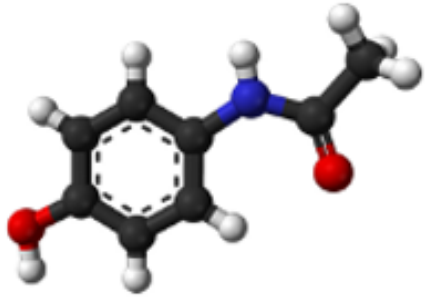


Message Passing



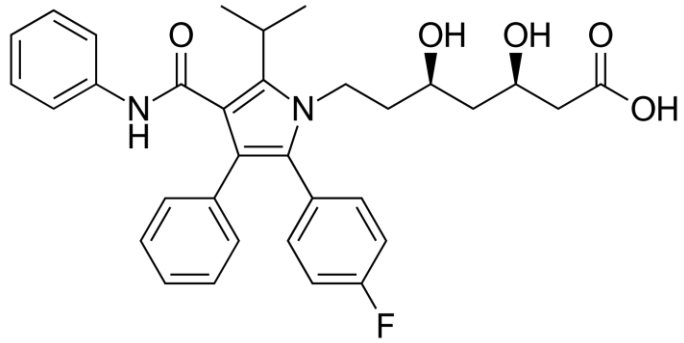
My research with GNN

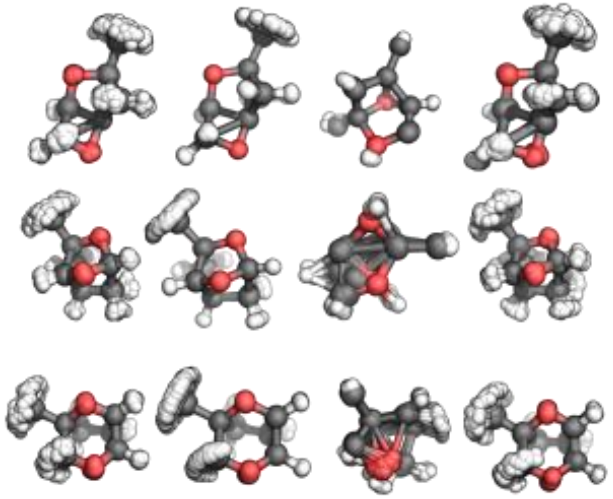
Predict molecular toxicity



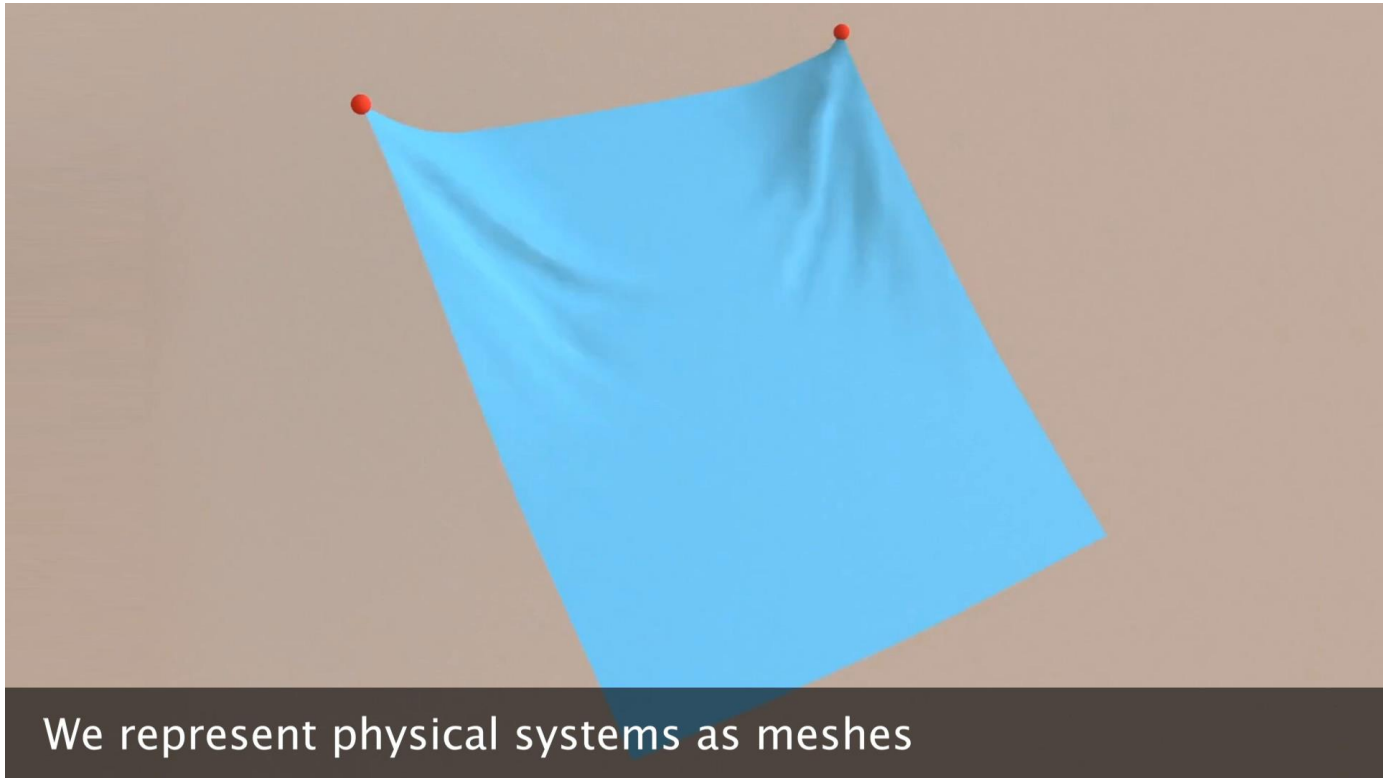
Toxic?

Geometry of molecule





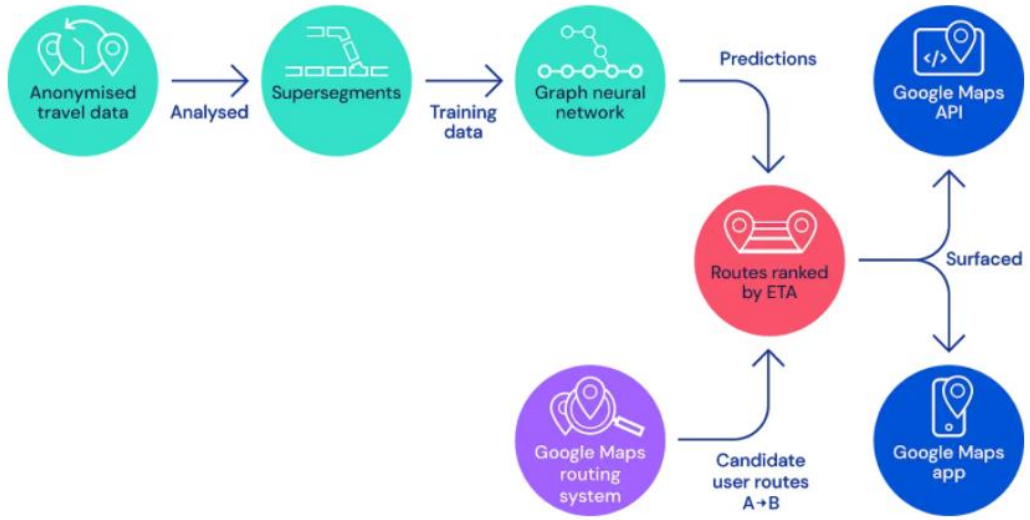
Mesh-based simulation

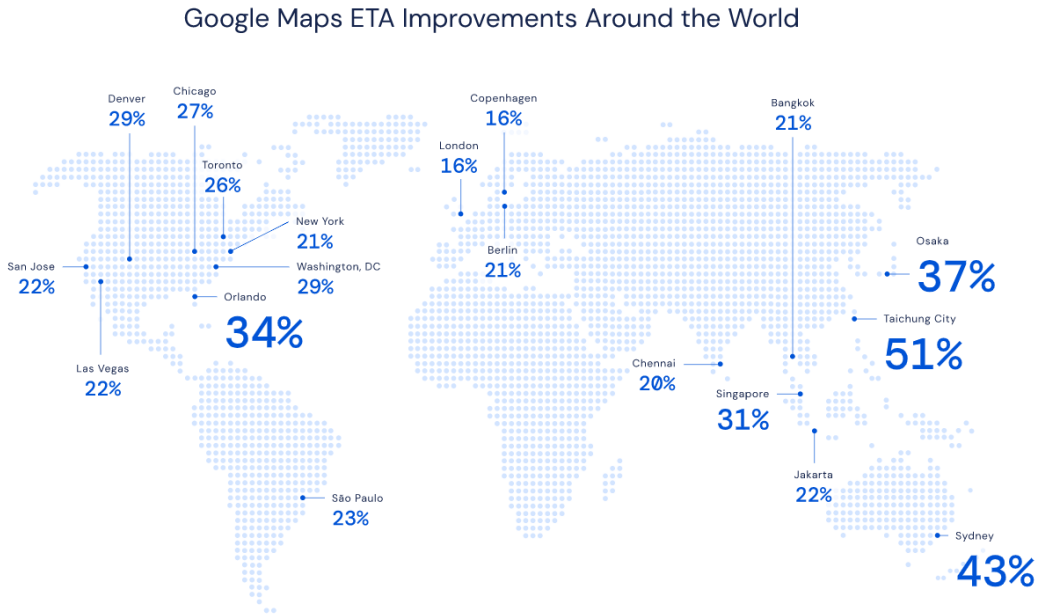


Learning mesh-based simulation with Graph Networks (Deepmind), https://arxiv.org/abs/2010.03409

Traffic predictions

Deep mind and google maps use Graph Neural networks to improve traffic predictions

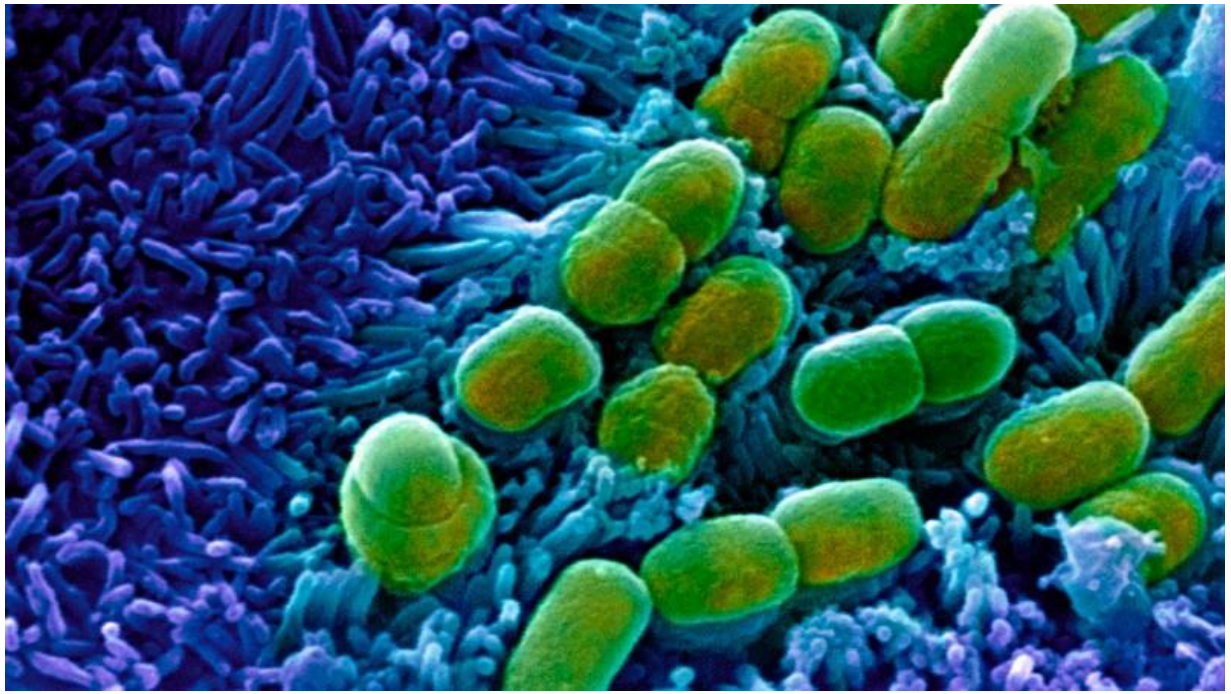




https://deepmind.com/blog/article/traffic-prediction-with-advanced-graph-neural-networks

Antibiotic Discovery

Stokes, *Jonathan* M., et al. "A deep learning approach to antibiotic discovery." *Cell*



They have, for the first time, identified completely new kinds of antibiotic from scratch, without using any previous human assumptions. Published in Cell.

Libraries

Pytorch

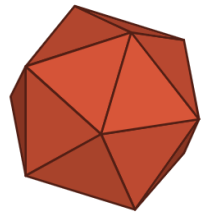
- Pytorch geometric
- Deep graph library

Tensorflow

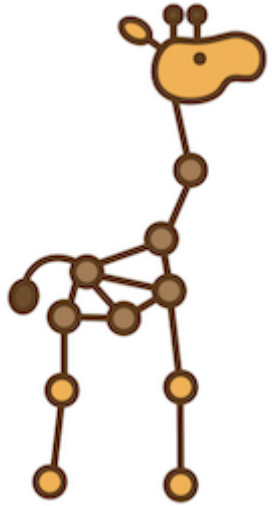
- Deep graph library
- Graph Nets library
- Spektral

Jax

- Jraph



PyTorch geometric



DEEP GRAPH LIBRARY

Spektral

Contact me: maria.bankestad@ri.se

References:

- About Non-Euclidean Spaces
- Bronstein, Michael M., et al. "Geometric deep learning: going beyond Euclidean data." IEEE Signal Processing Magazine 34.4 (2017): 18-42.
- Bacciu, D., Errica, F., Micheli, A., & Podda, M. (2020). A gentle introduction to deep learning for graphs. *Neural Networks*.
- Ward, I. R., Joyner, J., Lickfold, C., Rowe, S., Guo, Y., & Bennamoun, M. (2020). A Practical Guide to Graph Neural Networks. *arXiv preprint arXiv:2010.05234*.