Deep Learning on Graphs II:
Advanced Topics, Lecture 2

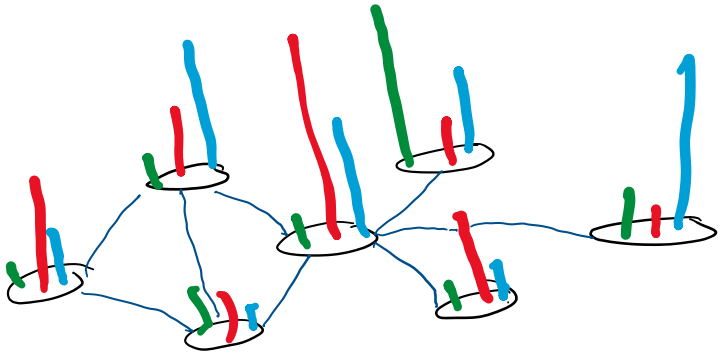
Michael Kenning, Stavros Georgousis

Outline

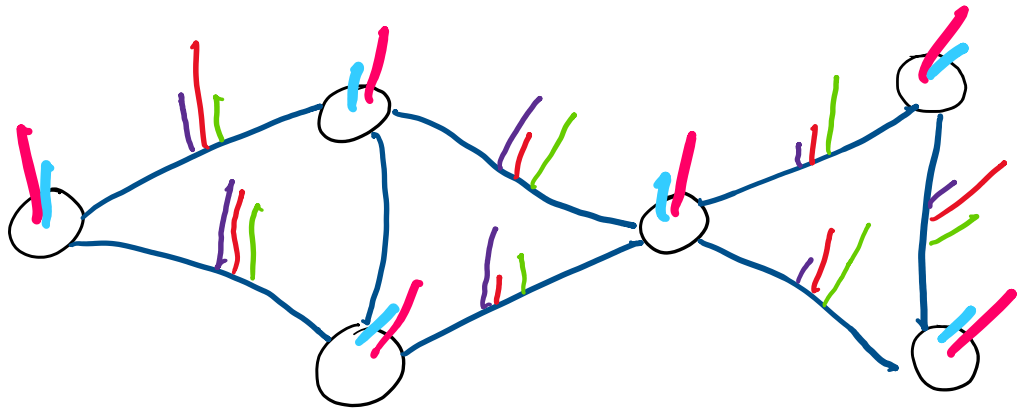
- Stavros' lecture:
  - Temporal graphs, and
  - Miscellaneous problems.
- Michael's lecture:
  - Edge attributes and signals, and
  - Graph estimation.

What Are Edge Signals/Attributes?

- Signals structured on vertices.

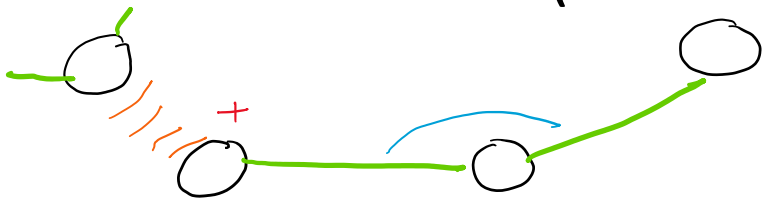


- Signals/attributes structured on the graph's edges.



Example: Molecules and Molecular Interactions

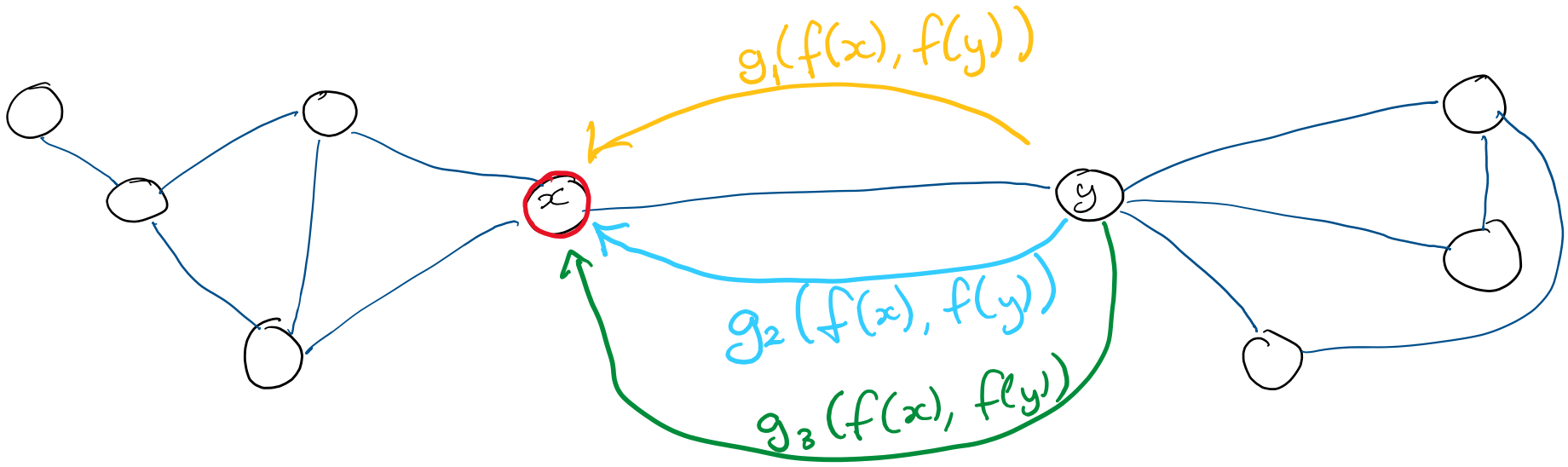
- Vertices as atoms (the *entities*), edges as chemical bonds (the *interactions/relations*).
- Represent information on vertices:
  - Valence, atomic number, mass, etc.
- **How** and **where** do we represent the following information?
  - Bonding distance (second-order interaction);
  - Bonding angles (third-order interaction);
  - Intramolecular bonding (second-order interaction)
  - Secondary structures in proteins (high-order interaction).



Edge Features: Not as Straightforward

Identified three situations where there are edge signals/attributes:

- Edge attributes that specify the interaction of the adjacent vertices.
  - *e.g.* knowledge graphs, or valence bonding, intra- and inter-molecular bonding, *etc.*
  - The edge attributes here may be discrete.



Example: Edge-conditioned Convolution (Simonovsky and Komodakis, 2017)

- The edges are mapped to a label:

$$L : E \rightarrow \mathbb{R}^S$$

- The label is used to index a weight matrix:

$$F : \mathbb{R}^S \rightarrow \mathbb{R}^{d \times c}$$

- The neighbours' signals are projected through the matrix:

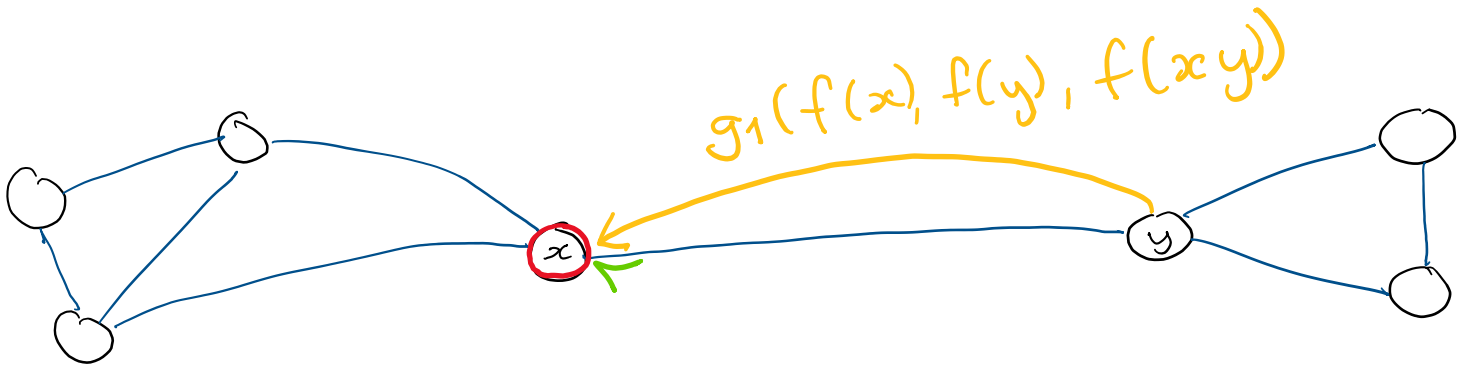
$$Z = \sum_{y \in \Gamma(x)} w_l \, F_l(L(xy)) \, f(x)$$

where l is the current layer's index and $w_l \in \mathbb{R}^d$ is the current layer's learned weights.

Edge Features: Not as Straightforward

Identified three situations where there are edge signals/attributes:

- Edge attributes that specify the interaction of the adjacent vertices.
  - *e.g.* knowledge graphs, or valence bonding, intra- and inter-molecular bonding, *etc.*
- Edge signals integrated into learning on the vertices
  - *e.g.* distances/angles of chemical bonding or intramolecular bonding
  - Must be continuous.



Example: Weaving (Kearnes *et al.*, 2016)

- One unit for the vertices/atoms:

$$(A \longrightarrow A) : f(A_{0,a}, A_{1,a}, \cdots, A_{l-1,a}) = A_{l,a} \quad \text{for atom } a$$

- One unit for the links/pairs of atoms:

$$(P \longrightarrow P) : f(P_{0,ab}, \cdots, P_{l-1,ab}) = P_{l,ab} \quad \text{for atom-pair } ab.$$

- Another two units that merge the output of the two:

The Weave modules

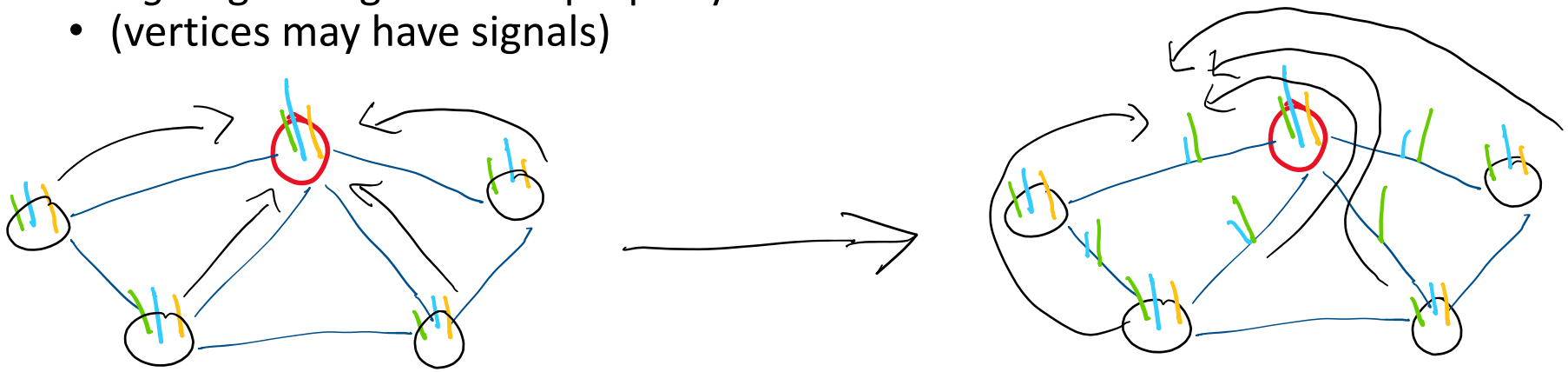
$$(P \longrightarrow A) : f(P_{l-1,ab}, P_{l-1,ac}, \cdots) = P_{l,a}$$

$$(A \longrightarrow P) : g(f(A_{l-1,a} A_{l-1,b}), f(A_{l-1,b}, A_{l-1,a})) = P_{l,ab}$$

Edge Features: Not as Straightforward

Identified three situations where there are edge signals/attributes:

- Edge attributes that specify the interaction of the adjacent vertices.
  - *e.g.* knowledge graphs, or valence bonding, intra- and inter-molecular bonding, *etc.*
- Edge signals integrated into learning on the vertices
  - *e.g.* distances/angles of chemical bonding or intramolecular bonding
- Learning on the edges themselves
  - *e.g.* regressing on some property
  - (vertices may have signals)



Example: Graph edge convolutional network (Zhang *et al.*, 2020)

- For each edge, aggregates its neighbourhood's signals:

$$\Gamma(e_{xy}) = \{ e_{wz} \mid e_{wk} \in E \wedge d(e_{wz}, e_{xy}) \leq 1 \} \quad \therefore \; e_{xy} \in \Gamma(e_{xy})$$

- Uses a labelling function l to select a weight matrix:

$$l : E \longrightarrow L \qquad w : L \longrightarrow \mathbb{R}^{d \times c}$$

- Sums (and normalises; not shown) the features:

$$f'(e_{xy}) = \sum_{e_{wz} \in \Gamma(e_{xy})} w(l(e_{wz}))\, f(e_{xy}).$$

Edge Signals and Attributes: Lacunae

- The particular approach is *ad hoc*:
  - No comparative analysis of the techniques of including edge signals in convolution.
- None considers a situation where the signals exist on the edges alone:
  - What does this mean? (See the next slide.)
- What about higher-order structures on the graph?
  - Not just bonding distances, but bonding angles.
  - How to represent higher-order interactions? (See the next slide, too.)

Higher-order Interactions: Linegraphs

- One solution is to model higher-order interactions with linegraphs.
- A linegraph represents the second-order structure of a graph.
  - The linegraph of a linegraph represents the third-order structure of a graph, *et cetera ad nauseum*.
- How to combine the information at each layer?

Linegraphs

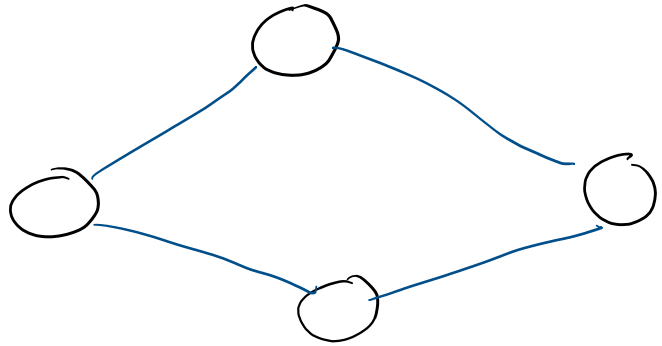
- Suppose we have a graph

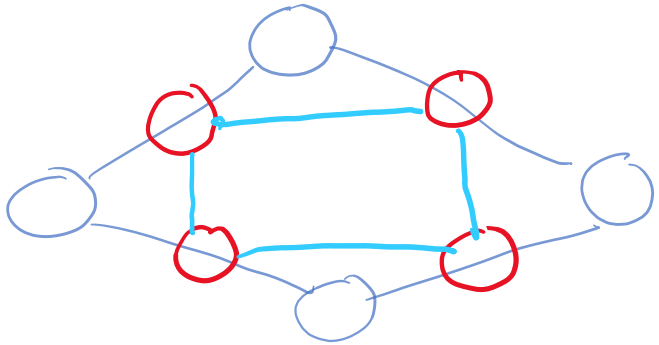
$$G = \langle V, E \rangle.$$

- A linegraph is

$$L(G) = \langle V_L, E_L \rangle = \langle E, E_L \rangle$$

- In other words—
  - The edges map to vertices in $L(G)$;
  - Two vertices in $L(G)$ are adjacent if their edges in G are adjacent.

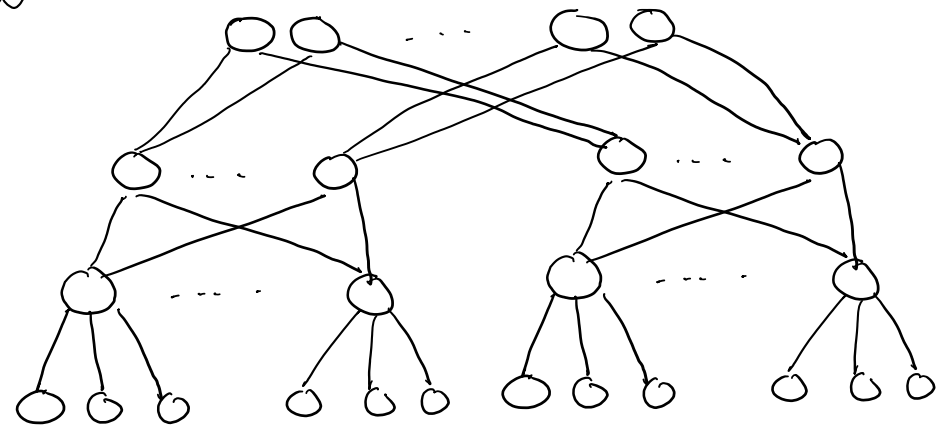




Example: Datacentres

- Link failures, where the signals exist only on the edges.
- Local effects, so structure is important.
- Represented well as a (line-)graph.
- Convolution works as on a normal graph.

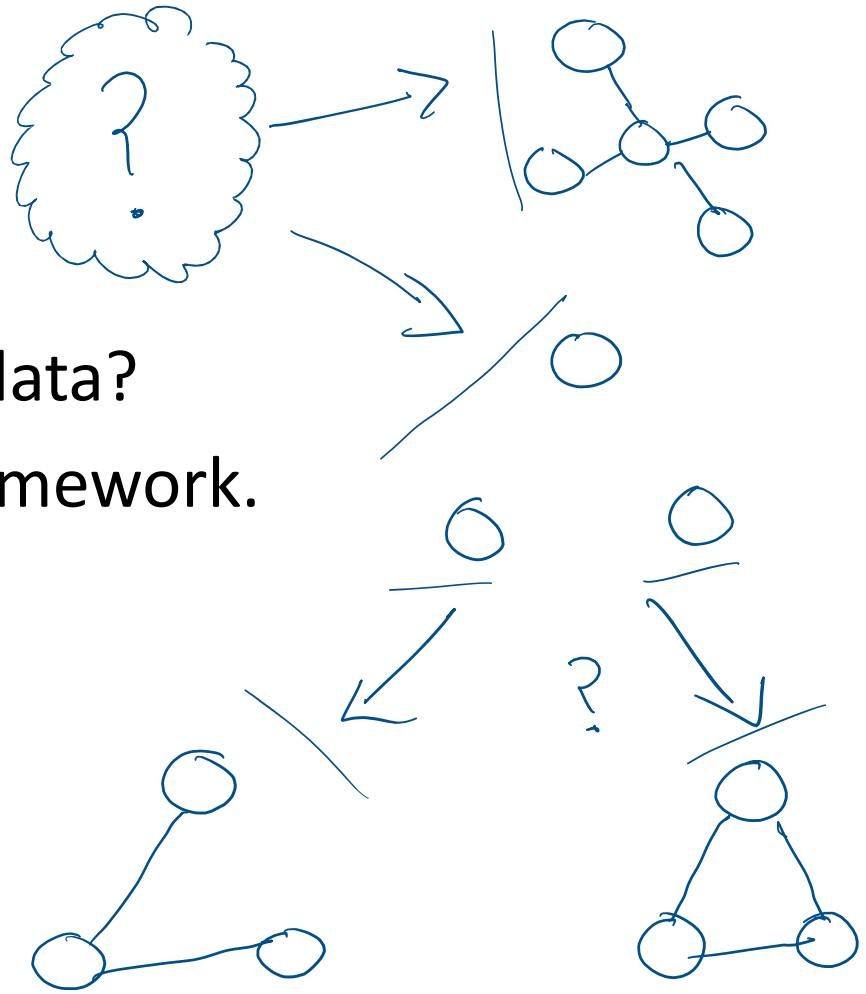
A datacentre.



The vertices represent machines, the edges represent connections.

In the linegraph, the links would be represented as vertices. Two links would be connected iff they have a machine in common.

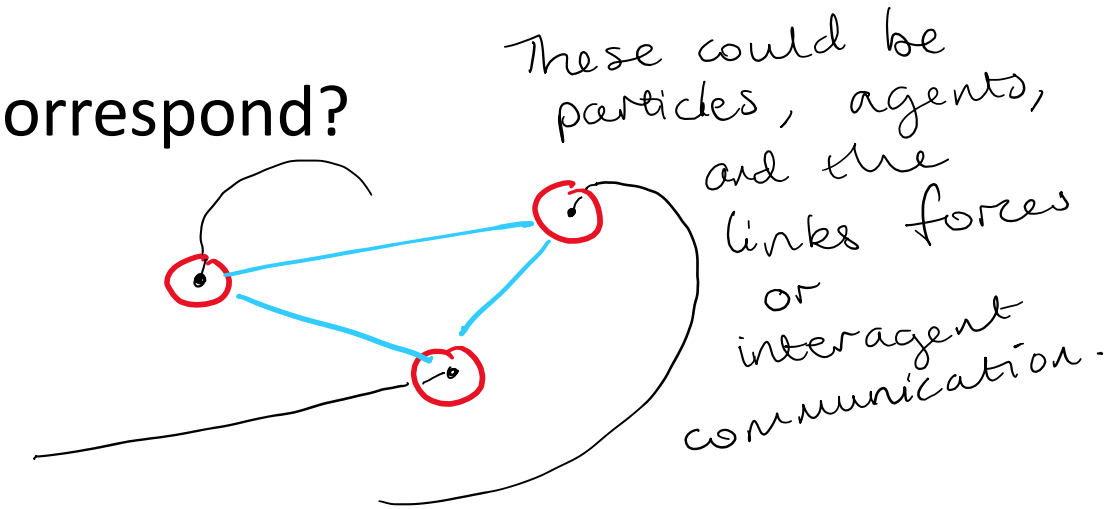
Graph Estimation



- Can one learn the graph structure present in data?
- Various terms in the literature; no unifying framework.
  - *Neural relational inference*
  - *Graph estimation*
  - *Graph generation*
  - *Structure learning*
- A question of missing structure.
- Some applications:
  - Learning the structure of an EEG scan, drug discovery, extrapolating the interactions in physical systems

How to Learn the Structure?

- To what real things does the structure correspond?
  - A graph describes—
    1. the *relations/interactions* between
    2. a set of *discrete entities.*
- Relations predicated on entities.
- How to discover entities?
  - If we don't know the entities—
    - Ontological and epistemological question, so not exactly easy.
    - As a shortcut, side-step the philosophical issue: take an arbitrary partitioning.
  - If we know the entities, the problem is more tractable.

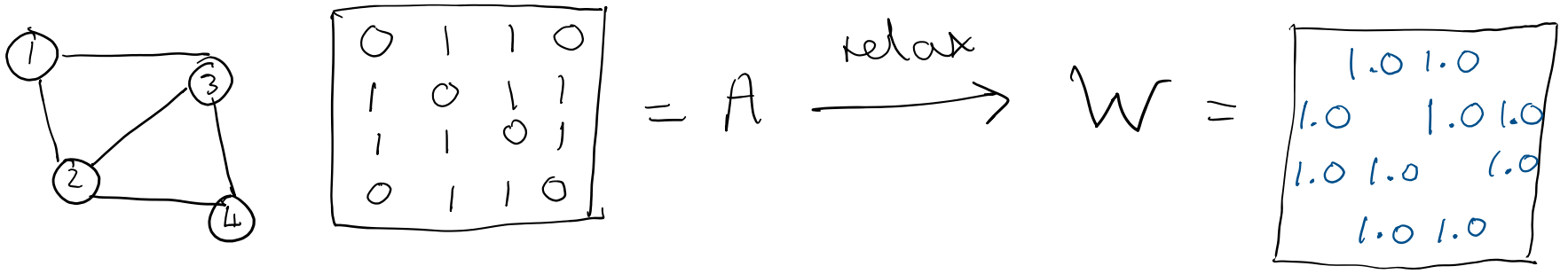


Learning the Relations of Entities: Challenges

- The search-space of an n-graph is enormous: 2^{n^2} solutions.
- Graphs are discrete, therefore non-differentiable.
- What kind of graph one wants changes the practicability of learning—and the complexity of the workspace.
- The initial graph may bias the learning in one direction or another.

Learning the Relations of Entities: Solutions

- Four ways identified in the literature:
  1. Continuous relaxation: learn the entries of the weight matrix end-to-end.

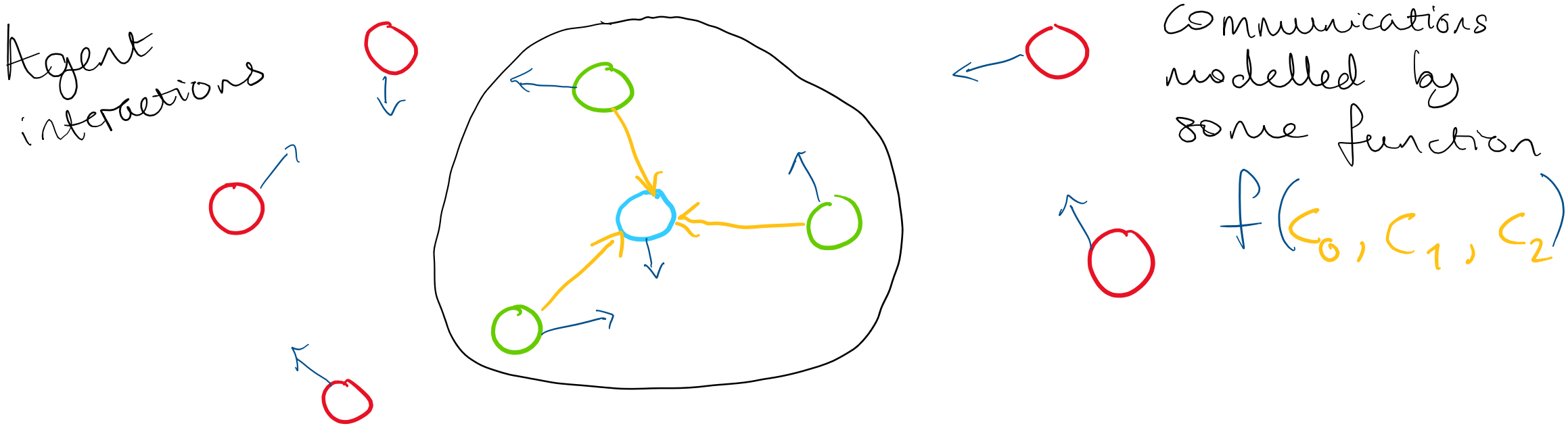


$\frac{dA}{dL}$ ✗

$\frac{dW}{dL}$ ✓

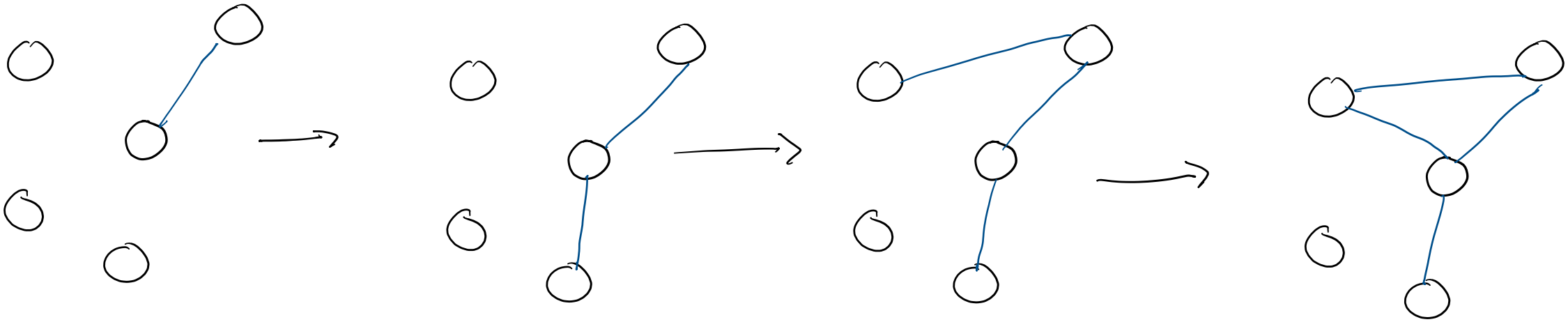
Learning the Relations of Entities: Solutions

- Four ways identified in the literature:
  1. Continuous relaxation: learn the entries of the weight matrix end-to-end.
  2. Learn the interaction function.



Learning the Relations of Entities: Solutions

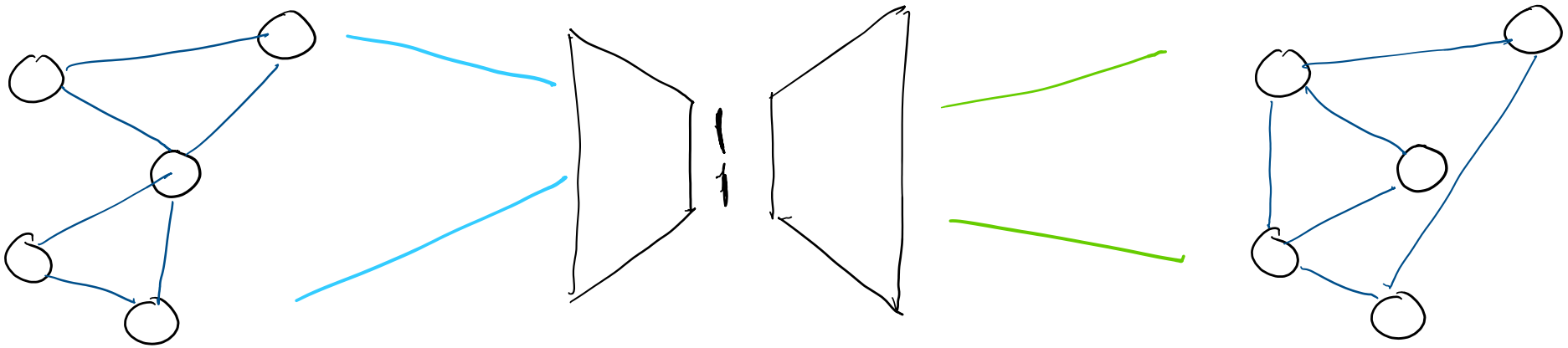
- Four ways identified in the literature:
  1. Continuous relaxation: learn the entries of the weight matrix end-to-end.
  2. Learn the interaction function.
  3. Generate edges sequentially.



Using recurrent units mostly.

Learning the Relations of Entities: Solutions

- Four ways identified in the literature:
  1. Continuous relaxation: learn the entries of the weight matrix end-to-end.
  2. Learn the interaction function.
  3. Generate edges sequentially.
  4. Generate whole graphs simultaneously.



References

S. Kearnes, K. McCloskey, M. Berndl, V. Pande and P. Riley, ‘Molecular graph convolutions: Moving beyond fingerprints’, J. Comput.-Aided Mol. Des., vol. 30, no. 8, pp. 595-608, Aug. 2016.

M. Simonovsky and N. Komodakis, ‘Dynamic edge-conditioned filters in convolutional neural networks on graphs’, *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3693-3702, Dec. 2017.

X. Zhang, C. Xu, X. Tian and D. Tao, ‘Graph edge convolutional neural networks for skeleton-based action recognition’, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3047-3060, Aug. 2020.

QUESTIONS, PLEASE