Graph Networks in (Deep) Reinforcement Learning

Thibault Lahire

Presentation

Friday, 10 January 2020

- n balls trapped in a box
- submitted to the laws of elastic collisions (and gravity)
- possibly interacting with each other !
- See video : `https://github.com/thibault-lahire/bouncing-balls-problem`

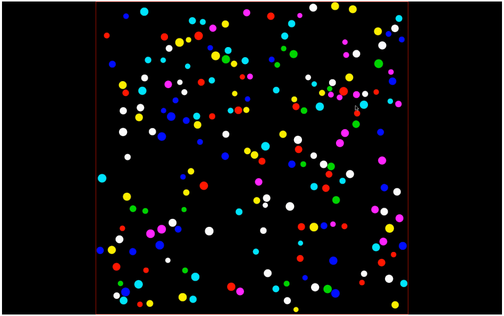

FIGURE – 200-bouncing-balls problem without interaction

- Definitions and properties of Graph Networks (GN)
- Illustration on the n-bouncing balls problem
- Illustration on the gluing task and links with Reinforcement Learning (RL)

Definitions

A **graph** G is defined by : $G = (V, E, \mathbf{u})$.
A **GN** is a function that takes a graph as input.

On the n-bouncing balls problem, it gives :

- V represents the balls, with attributes for position, velocity, radius, and mass.
- E represents the presence of springs between different balls, and their corresponding spring constants.
- $\mathbf{u}$ represents the total energy of the set of balls (other choices are also correct).

Two types of functions to consider :

- Update functions Φ
- Message passing or aggregation functions ρ

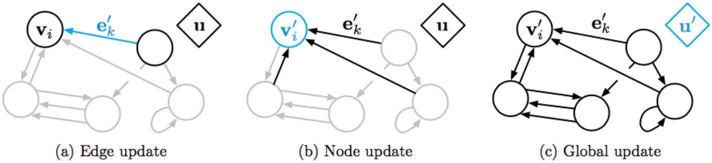

FIGURE – Updates in a GN block

Definition (simplified)

An inductive bias is a *prior* on the observed data.

Example : When doing image classification, using a CNN is a prior on the convolutional nature of images.

Definition

A **relational** inductive bias is a set of assumptions about the relational structure of the world.

Concretely : Using a GN to solve a problem of objects interacting with each other **is** a relational inductive bias.

Two main issues in "solving the problem of intelligence" !

Definition : combinatorial generalization

A problem is said **combinatorial** when solving it becomes difficult when the number of objects involved grows large.

Definition : learning transfer

The **learning transfer** consists in transferring the result of a training on a specific system to another one.

If a child already knows how to open a door, he/she doesn't have to make a lot of efforts to learn how to open a window...

No ! We have 2 cases to consider :

If the problem is *simple*

More precisely, as long as you can write $f(t+\Delta t) \simeq f(t) + \Delta t f'(t)$, you can predict the state of the system at time $t + \Delta t$ if you know everything at time t.

$\longrightarrow$ Graphs ARE NOT the best answer !

If the problem is *hard*

More precisely, when writing all the equations the system has to follow is tedious, or when equations are simply unknown :

$\longrightarrow$ Graphs ARE the best answer !

Consider there is no interaction between balls

- The ball movement is rectilinear if no gravity
- Collisions are our main concern : use an *event-driven simulation*.
- Store the events in a binary heap that ensures a complexity in $O(\log n)$ for storing and accessing the events.
- And we're done!

But imagine something a bit more complicated... interactions (springs) between balls... and a graph structure appears!

<u>When balls are connected, edges appear between vertices</u>
$\longrightarrow$ We would like to have a structure that takes graphs as input : use GN ! ! !

- You can *write by hand* all equations involving a node, and thus hard-coding the functions Φ and ρ...
- But you can also use deep neural networks as Φ- and ρ-approximators !

So we train the networks on the trajectories from time $\tau = 1$ to time $\tau = t$, and the goal is to predict the system at time $t + 1$.

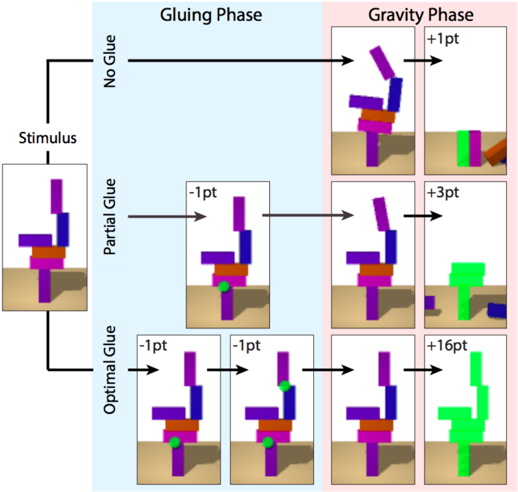

FIGURE – The gluing task

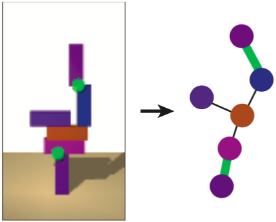

FIGURE – The problem seen as a graph

There are 5 agents :

- Humans : they look at the tower
- MLP and GN-FC : they are given a fully connected graph as input $\rightarrow N(N-1)/2$ possible edges to glue
- GN : it is given a graph where the edges encode the contact between blocks $\rightarrow \simeq N$ possible edges to glue
- Sim : a reference baseline which is able to compute the forces when gravity will be applied

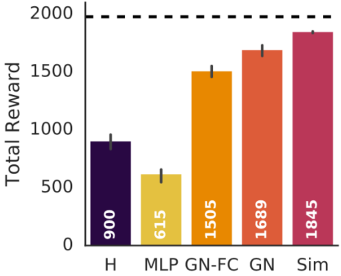

FIGURE – Comparison between the 5 agents

Conclusion

Two important challenges for the future :

- Combinatorial generalization
- Learning transfer

Two great ideas we have to tackle these issues :

- Relational inductive biases
- Graphs Networks

References

[1] Battaglia, P. et al. 'Interaction Networks for Learning about Objects, Relations and Physics' (https://arxiv.org/abs/1612.00222)

[2] Sanchez-Gonzalez, A. et al. 'Graph networks as learnable physics engines for inference and control' (https://arxiv.org/pdf/1806.01242.pdf)

[3] B. Hamrick, J. et al. 'Relational inductive bias for physical construction in humans and machines' (https://arxiv.org/abs/1806.01203)

[4] `https://gym.openai.com/envs/#classic_control`

[5] Scarselli, F., Gori, M., Tsoi, A., Hagenbuchner, M. and Monfardini, G. 2009, 'The graph neural network model', IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61-80 : `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1015.7227&rep=rep1&type=pdf`

[6] Battaglia, P. et al. 'Relational inductive biases, deep learning, and graph networks' `https://arxiv.org/pdf/1806.01261.pdf`

[7] Kansky, K., et al. (2017). Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. In Proceedings of the International Conference on Machine Learning (ICML).

[8] Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., and Tenenbaum, J. B. (2010). Probabilistic models of cognition: exploring representations and inductive biases. TiCS, 14, 357–364.

[9] Spelke, E. S., and Kinzler, K. D. (2007). Core knowledge. Develop- mental Science, 10, 89–96.

[10] Kipf T., et al., 'Neural Relational Inference for Interacting Systems' `https://arxiv.org/abs/1802.04687`

[11] Kingma D., et al., 'Auto-Encoding Variational Bayes' `https://arxiv.org/abs/1312.6114`