

Redes neurais de grafos

Douglas De Rizzo Meneghetti

14 de outubro de 2019

Centro Universitário FEI

Programa de Doutorado em Engenharia Elétrica

Inteligência Artificial Aplicada à Automação e Robótica

Introdução

Grafos

Modelagem de Dados Utilizando Grafos

Redes Neurais de Grafos

Aprendizado por Reforço utilizando GNNs

Conclusão

Introdução

As redes neurais propostas nos últimos tempos possuem, incorporadas em suas arquiteturas, vieses que as tornam especializadas em trabalhar com dados de naturezas específicas.

As redes neurais propostas nos últimos tempos possuem, incorporadas em suas arquiteturas, vieses que as tornam especializadas em trabalhar com dados de naturezas específicas.

Redes convolucionais preservam as relações espaciais dos pixels das imagens nos filtros de convolução.

Redes recorrentes modelam em vetores ocultos o histórico de dados sequenciais visualizados.

Essas arquiteturas de redes neurais possuem vantagens decorrentes da especialização no tipo de dados de entrada, como:

- compartilhamento de parâmetros
- geração hierárquica de características

Essas arquiteturas de redes neurais possuem vantagens decorrentes da especialização no tipo de dados de entrada, como:

- compartilhamento de parâmetros
- geração hierárquica de características

Porém, existem dados que podem ser modelados de forma mais natural através de *grafos*.

Aprendizado profundo geométrico é o nome da área que estuda a criação e aplicação de técnicas de aprendizado de máquina em dados que são de natureza não-Euclideana¹.

¹<http://geometricdeeplearning.com/>

Aprendizado profundo geométrico é o nome da área que estuda a criação e aplicação de técnicas de aprendizado de máquina em dados que são de natureza não-Euclideana¹.

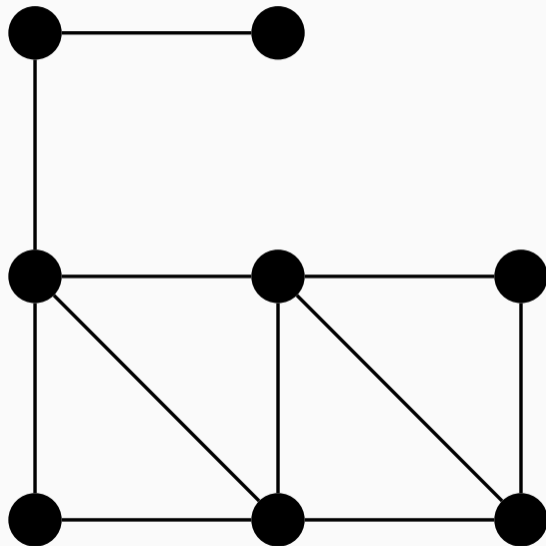
Uma forma comum de se organizar dados desta natureza é através do uso de *grafos*.

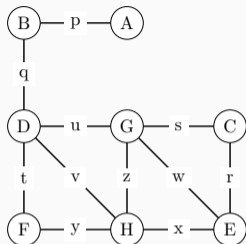
¹<http://geometricdeeplearning.com/>

Grafos

Um grafo G é composto por um conjunto não-vazio de vértices (nós) $V(G)$, um conjunto de arestas $E(G)$ e uma função de incidência ψ_G , a qual relaciona arestas de $E(G)$ com pares de vértices (não necessariamente distintos) de $V(G)$ [BM08].

Exemplo Gráfico de um Grafo





$$V(G) = A, B, C, D, E, F, G, H$$

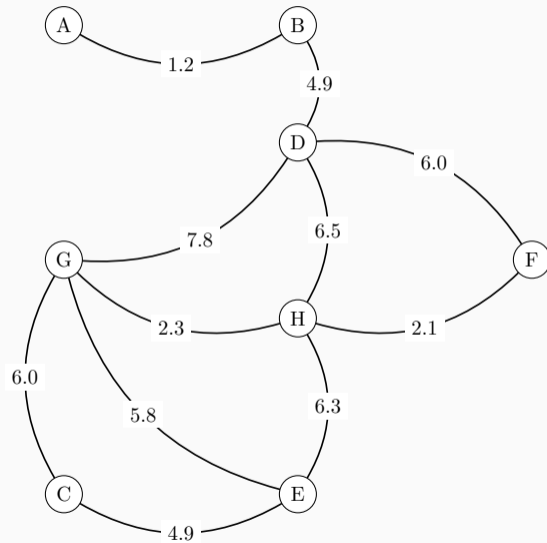
$$E(G) = p, q, r, s, t, u, v, w, x, y, z$$

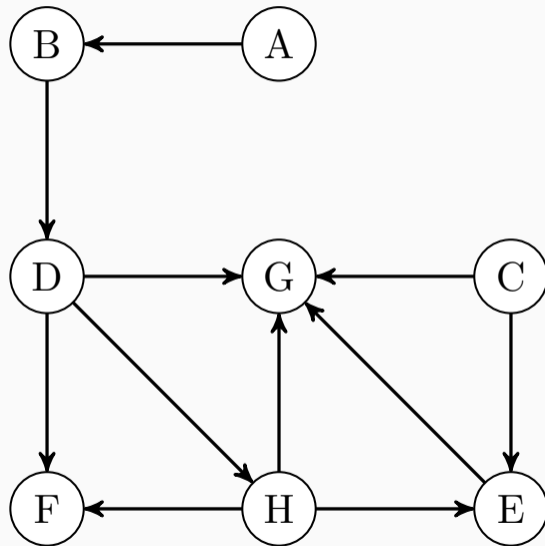
$$\psi_G(p) = AB \quad \psi_G(q) = BD \quad \psi_G(r) = CE \quad \psi_G(s) = CG$$

$$\psi_G(t) = DF \quad \psi_G(u) = DG \quad \psi_G(v) = DH \quad \psi_G(w) = EG$$

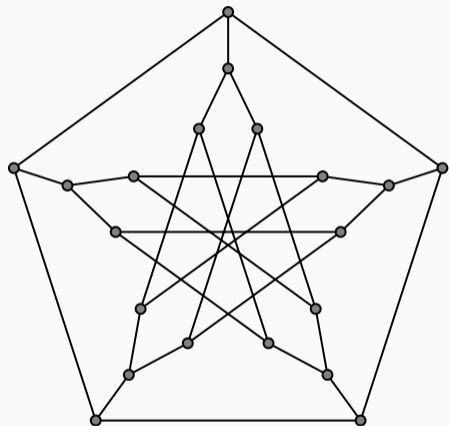
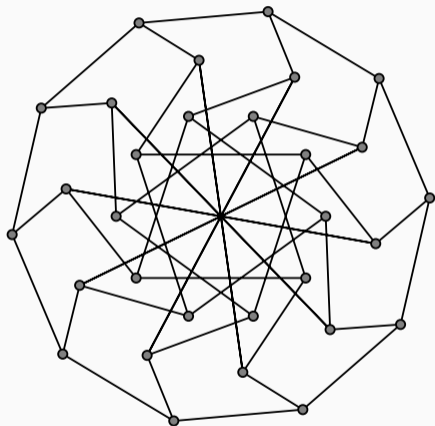
$$\psi_G(x) = EH \quad \psi_G(y) = FH \quad \psi_G(z) = GH$$

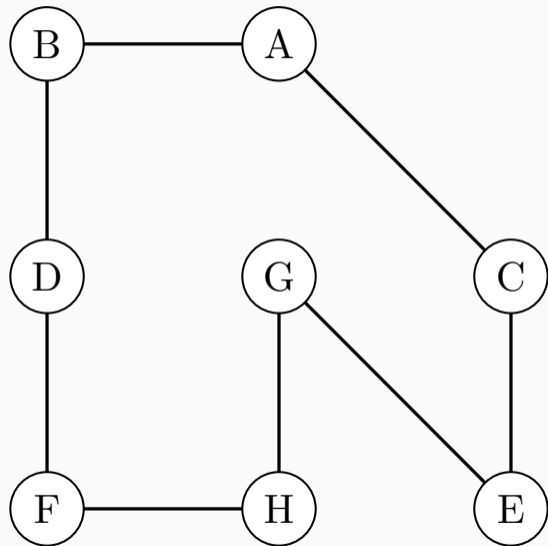
Grafo Ponderado

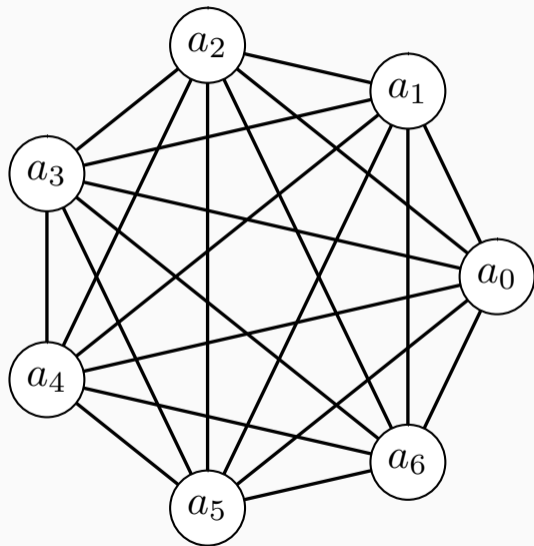




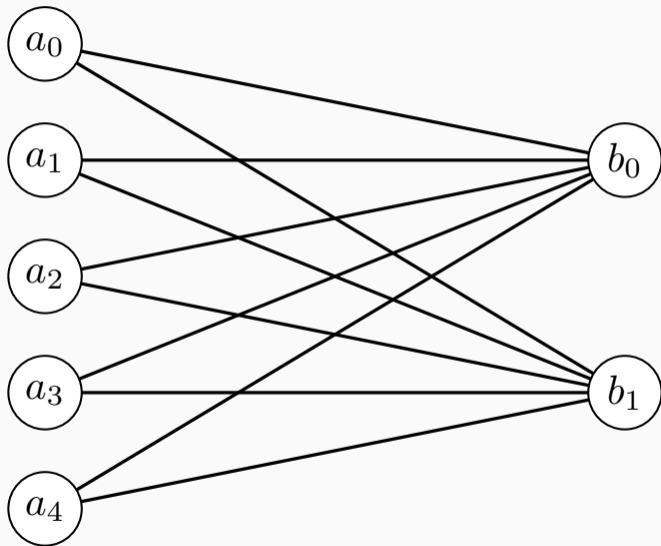
Grafos Regulares

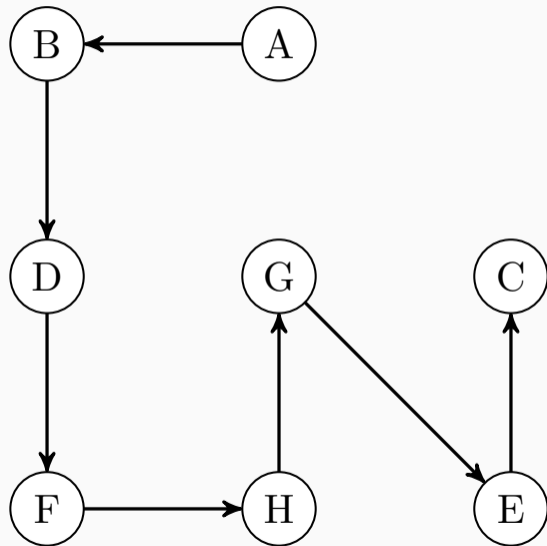






Grafo Bipartido Completo $K_{5,2}$





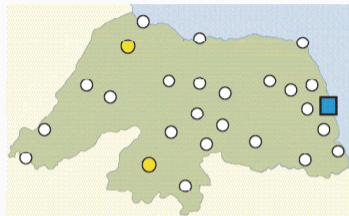
Modelagem de Dados Utilizando Grafos

Alguns tipos de dados podem ser naturalmente representados como grafos...

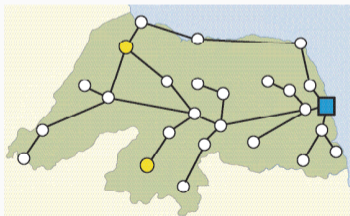
Mapas



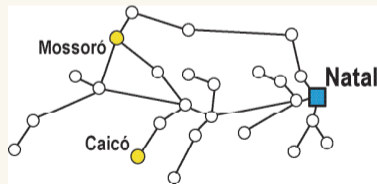
(1) Mapa do Rio Grande do Norte



(2) Localização das cidades



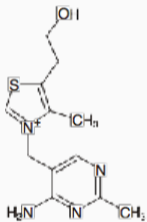
(3) Ligação entre as cidades do Estado



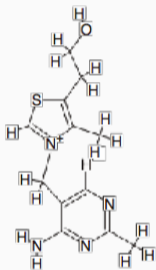
(4) Grafo associado

Structure: Thiamine (Vitamin B₁)

molecular graph

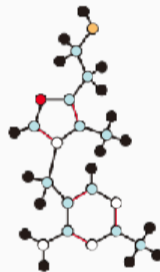


implicit
hydrogens



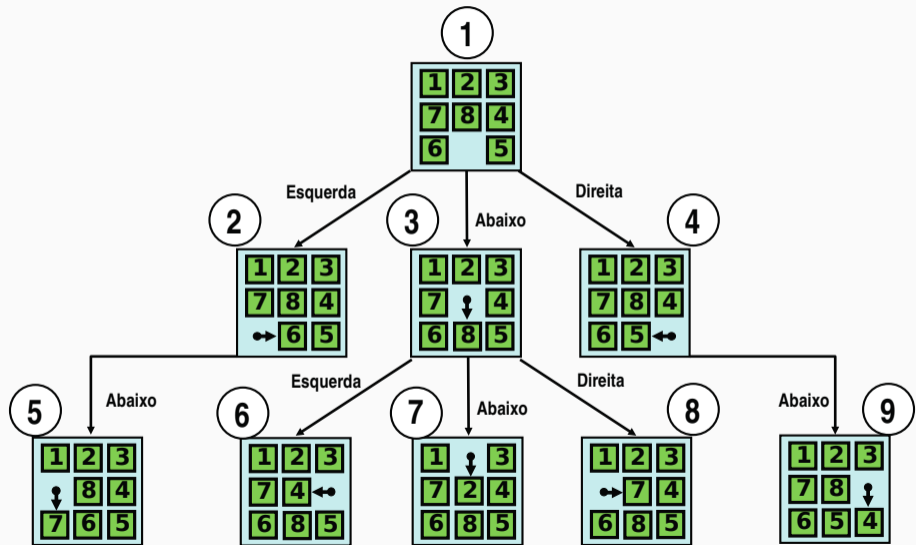
explicit
hydrogens

abstraction

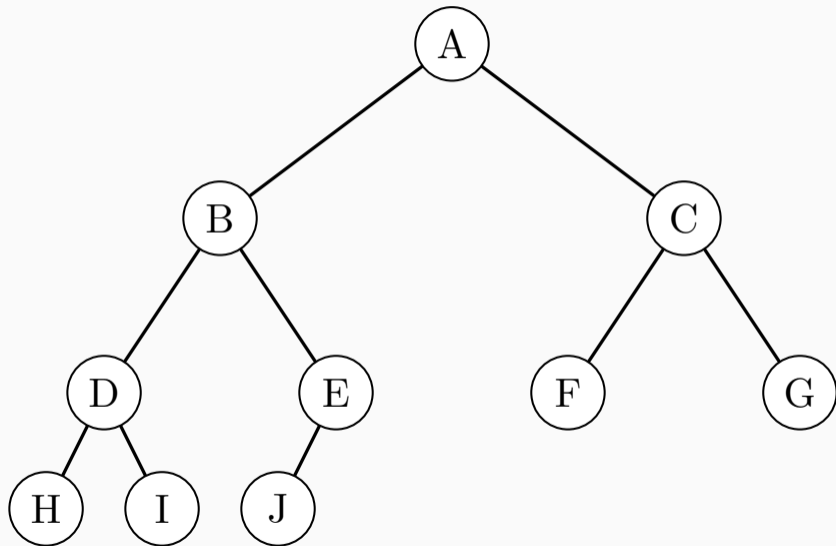


- Hydrogen
- Carbon
- Oxygen
- Nitrogen
- Sulfur
- single bond
- double bond

Modelagem de estados



Modelagem de outras estruturas de dados



Outros tipos de dados não possuem representações diretas para grafos. Mas algumas soluções existem!

- Criar um grafo usando uma heurística (e.g. kNN)
- Aprender a representação junto da tarefa

- Como descobrir que uma nova molécula é uma droga, dados os grafos moleculares de outras drogas e de moléculas que não são drogas?
- Como descobrir as condições climáticas de uma cidade, dadas as condições climáticas de cidades vizinhas?

Utilizando aprendizado supervisionado, como descobrir:

- uma propriedade de um novo grafo, dada uma base de grafos com propriedades conhecidas?
- uma propriedade desconhecida de um nó em um grafo, dados os nós existentes em um ou mais grafos cujas propriedades são conhecidas?

Redes Neurais de Grafos

Propostas em [GMS05; Sca+09], com o objetivo de aprender, através de exemplos, uma função que mapeasse tanto um grafo G a um vetor de números reais, como um nó v .

- $\tau(G) = \mathbb{R}^m$
- $\tau(G, v) = \mathbb{R}^m$

Cada nó $v \in V$ possui um vetor de características I_v .

$x_v = f_{\mathbf{w}}(I_v, \mathbf{x}_{ne[v]}, I_{ne[v]})$ é um vetor de características de v que depende de se seus vizinhos, $ne[v]$.

$o_v = g_{\mathbf{w}}(I_v, x_v)$ é a saída de v .

$f_{\mathbf{w}}$ e $g_{\mathbf{w}}$ são funções parametrizadas por pesos \mathbf{w} . Na prática, f e g são redes neurais, MLPs ou RNNs parametrizadas por conjuntos de pesos diferentes (\mathbf{w}_f e \mathbf{w}_g).

Objetivo das GNN

Encontrar \mathbf{w} de forma a aproximar \mathbf{o}_v de uma saída esperada.

A base de exemplos é composta por p triplas (G, v, t) , onde G é um grafo, v é um vértice de G e t é a saída esperada para uma função $\phi_{\mathbf{w}}(G_i, v_i) = \mathbf{o}_v$.

O valor de \mathbf{w} é encontrado minimizando uma função de erro, como o erro quadrático:

$$e_{\mathbf{w}} = \sum_{i=1}^p (t_i - \phi_{\mathbf{w}}(G_i, v_i))^2$$

Passo 1: Estabilização de x_v

- x_v depende de $x_{ne[v]}$.
- Necessário atualizar iterativamente $x_v(t) = f_w(l_v, x_{ne[v]}(t-1), l_{ne[v]})$ até alcançar um ponto fixo estável em $t = T$.

Passo 1: Estabilização de x_v

- x_v depende de $x_{ne[v]}$.
- Necessário atualizar iterativamente $x_v(t) = f_{\mathbf{w}}(I_v, \mathbf{x}_{ne[v]}(t-1), I_{ne[v]})$ até alcançar um ponto fixo estável em $t = T$.

Passo 2: cálculo dos pesos

- Calcular o gradiente $\frac{\partial e_{\mathbf{w}}(T)}{\partial \mathbf{w}}$ e atualizar \mathbf{w} usando descida de gradiente.

- nós representam objetos ou conceitos, descritos por um vetor de características.
- arestas representam relações (estradas, ligações moleculares).
- a GNN otimiza seu desempenho em uma tarefa (e.g. classificação, regressão) modelando as interações entre os objetos em seus pesos w .

Compartilhamento de parâmetros

Ambas as funções f_w e g_w são aplicadas a todos os nós dos grafos, o que significa que w é utilizada em mais de um local na entrada dos dados.

Compartilhamento de parâmetros

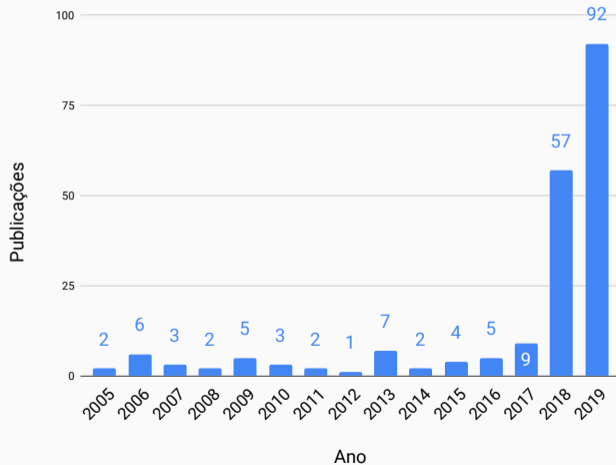
Ambas as funções f_w e g_w são aplicadas a todos os nós dos grafos, o que significa que w é utilizada em mais de um local na entrada dos dados.

Grafos de tamanho arbitrário

A ordem na qual $ne[v]$ é utilizado em f_w importa? Se não, f_w pode ser uma *função de agregação* (e.g. máximo, média), ignorando a ordem e quantidade de vértices.

Permite processar grafos de tamanhos arbitrários!

Termo “*graph neural network*” pesquisado no Scopus e IEEE Xplore em 09/2019.



Bibliotecas

- Deep Graph Library (backends em PyTorch e MXNet) – <https://www.dgl.ai/>
- Graph Nets (DeepMind) – https://github.com/deepmind/graph_nets
- PyTorch Geometric –
<https://pytorch-geometric.readthedocs.io/en/latest/>

Exemplos de redes em outros *frameworks*

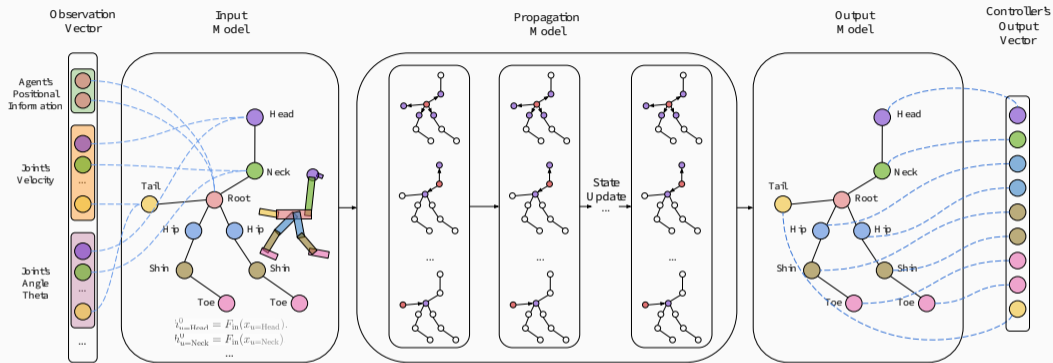
- <https://github.com/microsoft/tf-gnn-samples>
- [Li+16] –
<https://github.com/microsoft/gated-graph-neural-network-samples>

Aprendizado por Reforço utilizando GNNs

GNN criada para aprender a política de controle contínuo de cada articulação do modelo tridimensional de um animal no simulador de física MuJoCo [TET12].

Também são apresentados resultados para transferência da política aprendida em um esqueleto para outro.

Função de recompensa $R \leftarrow$ distância percorrida pelo animal simulado.



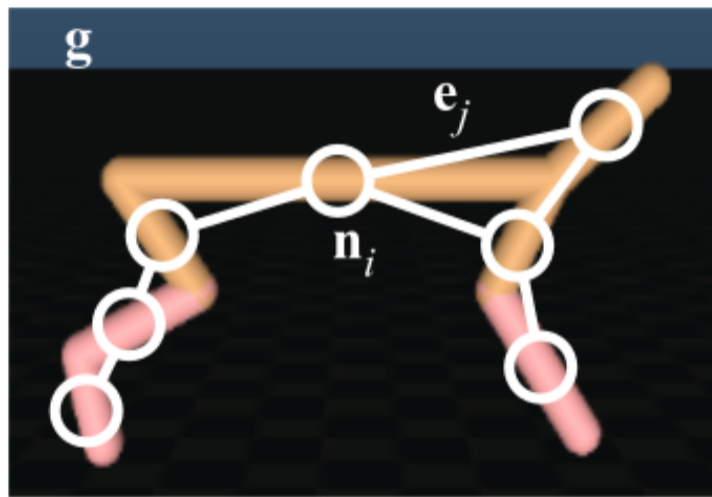
<https://www.youtube.com/watch?v=ImSlirW1EI8>

Graph Networks as Learnable Physics Engines for Inference and Control [San+18]

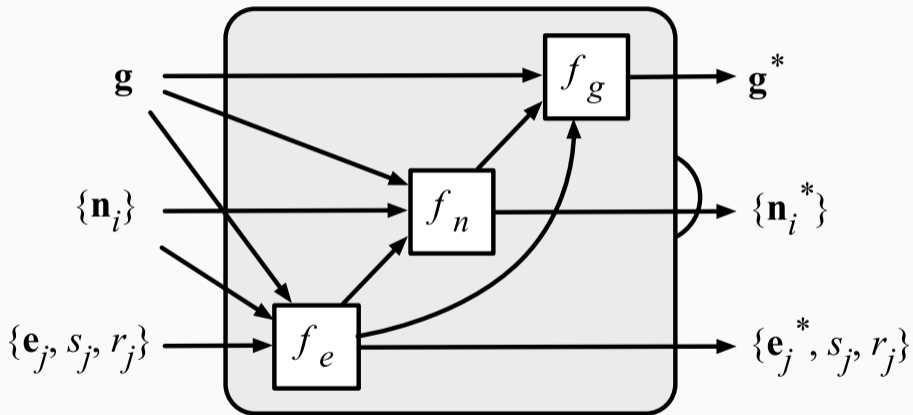
Realização de controle contínuo no MuJoCo e num braço robótico JACO.

Utiliza um "bloco" na rede neural que possui grafos como entrada e saída [Bat+18].

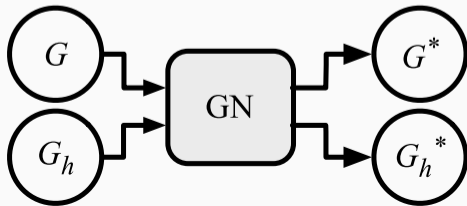
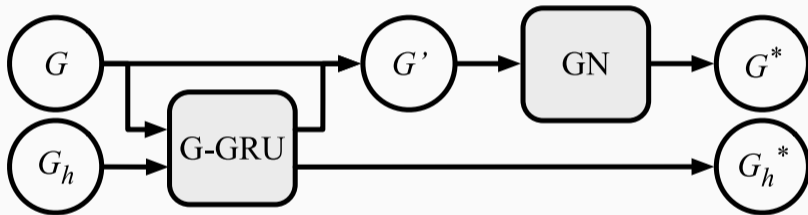
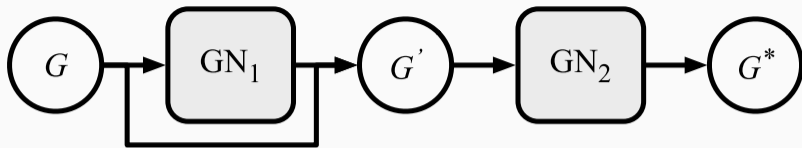
Exemplo de um modelo do MuJoCo.



"Bloco" que tem como entrada um grafo e como saída, outro grafo.



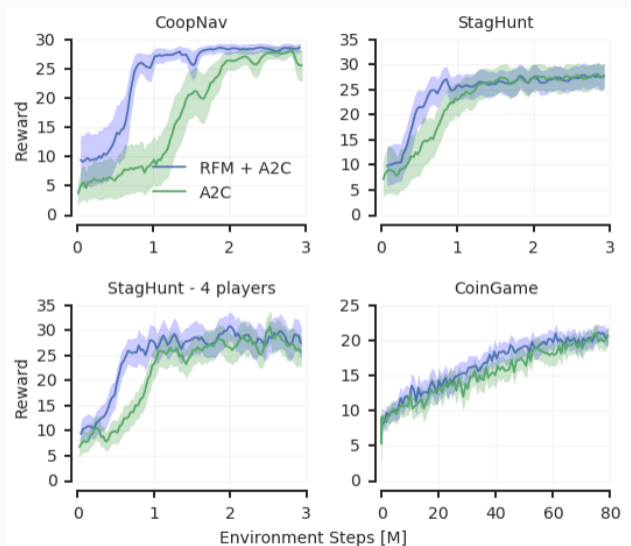
Diferentes propostas para se modelar as mudanças temporais em um grafo.



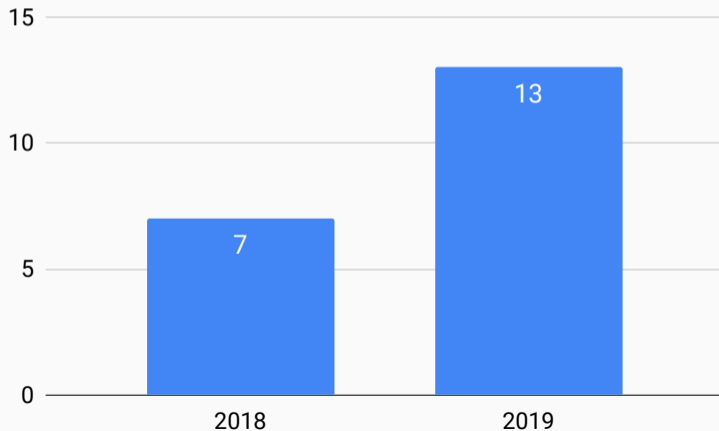
Criação de um modelo para predição de ações futuras de um sistema multi-agentes.

1. Agentes são treinados usando A2C
2. Um base de dados de 500 mil transições é salva
3. Os estados das transições são convertidos para grafos
4. Uma GNN é treinada para prever a ação que o agente toma, dado um grafo de entrada representando um estado

O modelo preditivo é então utilizado como parte da entrada de um novo agente A2C.



Termos “*graph neural network*” e “*reinforcement learning*” pesquisados no Scopus, IEEE Xplore e arXiv em 09/2019.



Conclusão

- A pesquisa em **aprendizado profundo geométrico** e **redes neurais de grafos** é uma área contemporânea de pesquisa
- Com as GNN, é possível realizar *classificação* e *regressão* de nós, arestas e grafos
- *Aplicações em diversas áreas* estão sendo descobertas (química, medicina, controle)
- *Diversos desafios* de pesquisa ainda existem, como o aprendizado de representações de dados em forma de grafos e aplicação em novas áreas

Referências

- [Bat+18] Peter W. Battaglia et al. “Relational inductive biases, deep learning, and graph networks”. Em: *arXiv e-prints*, arXiv:1806.01261 (4 de jun. de 2018). arXiv: 1806.01261 [cs.LG].
- [BM08] J. A. Bondy e U. S. R. Murty. *Graph Theory*. Springer London, 2008. DOI: 10.1007/978-1-84628-970-5.

- [GMS05] M. Gori, G. Monfardini e F. Scarselli. “A new model for learning in graph domains”. English. Em: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005*. Vol. 2. cited By 70. IEEE, jul. de 2005, pp. 729–734. DOI: 10.1109/ijcnn.2005.1555942.
- [Li+16] Yujia Li et al. “Gated graph sequence neural networks”. English. Em: cited By 86. International Conference on Learning Representations, ICLR, 2016. arXiv: 1511.05493 [cs.LG].
- [San+18] Alvaro Sanchez-Gonzalez et al. “Graph networks as learnable physics engines for inference and control”. Em: (4 de jun. de 2018). arXiv: 1806.01242v1 [cs.LG].

- [Sca+09] F. Scarselli et al. “The Graph Neural Network Model”. English. Em: *IEEE Transactions on Neural Networks* 20.1 (jan. de 2009). cited By 269, pp. 61–80. ISSN: 1045-9227. DOI: 10.1109/tnn.2008.2005605.
- [Tac+18] Andrea Tacchetti et al. “Relational Forward Models for Multi-Agent Learning”. Em: (28 de set. de 2018). arXiv: 1809.11044 [cs, stat].
- [TET12] Emanuel Todorov, Tom Erez e Yuval Tassa. “MuJoCo: A physics engine for model-based control”. Em: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, out. de 2012. DOI: 10.1109/iros.2012.6386109.

- [Wan+18] Tingwu Wang et al. “Nervenet: Learning structured policy with graph neural networks”. English. Em: *International Conference on Learning Representations*. cited By 8. International Conference on Learning Representations, ICLR, 2018.