GNNs in Physics
03/05/2024

AI
∩
Universe

L = -\frac{1}{2} \sum_{\mu} \tilde{D}_\mu \psi \tilde{D}^\mu \psi + h.c.

\tilde{D}_\mu \psi = \left( \frac{\partial}{\partial x^\mu} + \tilde{A}^\mu \right) \psi

\sum_{\mu} \tilde{D}_\mu \tilde{D}^\mu = \Delta + m^2

\tilde{A}_\mu^\dagger \tilde{A}_\mu + h.c.

\psi = \phi^* \nabla \phi - \nabla V(\phi)
A lot of data live on grids.

- **Image**: (HxWxC)
- **Video**: (TxCxHxW)
- **Text**: (N)-dim sequence
- **Speech**: (N)-dim sequence

"I love DL"
How about these:

Knowledge Graphs  Computer Networks  Disease Pathways

Food Webs  Molecules  Underground Networks

https://snap.stanford.edu/class/cs224w-2023/slides/01-intro.pdf
Social Networks  Economic Networks  Communication Networks

Citation Networks  Internet  Networks of Neurons

Image credit: Medium  Image credit: Science  Image credit: Lumen Learning

Image credit: Missoula Current News  Image credit: The Conversation

https://snap.stanford.edu/class/cs224w-2023/slides/01-intro.pdf
Data doesn’t always have a fixed structure. However, all those examples can be represented as graphs!
What is a Graph?

**Graph** - is a pair \((V,E)\), where \(V\) - is a set whose elements are called *vertices* and \(E\) is a set of (un)ordered pairs of vertices \(\{v1, v2\}\), whose elements are called *edges*. (Occasionally, this definition is being modified to include a *general feature* of the graph. In that case, graph is defined as a three-tuple \((V,E,u)\). )

![Directed and undirected fully-connected graphs](Im. Source: Wikipedia)
ML with Graphs

The key objective is to provide a framework to learn (and predict) from graph-represented data.

Examples include:

- node-level prediction (eg: predict a property of a given node (vertex))
- edge-level prediction (predict the connection bw. two given nodes)
- graph classification (predict a general property of a graph)
- graph generation
Message Passing Framework
(Relational inductive biases, deep learning, and graph networks, https://arxiv.org/abs/1806.01261)

Message Passing Graph Neural Nets is a general class of architectures for learning from graph-represented data.

(a) Edge update  (b) Node update  (c) Global update
Functions \( \{ \phi_e, \phi_n, \phi_u \} \) parametrize messages.

Permutation-invariant aggregation functions \( \rho \) aggregate computed messages to get updated feature embeddings.

Algorithm 1: Steps of computation in a full GN block.

```
function GRAPHNETWORK(E, V, u)
    for \( k \in \{1 \ldots N^e\} \) do
        \( e'_k \leftarrow \phi^e(e_k, v_{r_k}, v_{s_k}, u) \) \( \triangleright 1. \) Compute updated edge attributes
    end for
    for \( i \in \{1 \ldots N^n\} \) do
        let \( E'_i = \{(e'_k, r_k, s_k)\}_{r_k=i, k=1:N^e} \)
        \( e'_i \leftarrow \rho^{e \rightarrow u}(E'_i) \) \( \triangleright 2. \) Aggregate edge attributes per node
        \( v'_i \leftarrow \phi^v(e'_i, v_i, u) \) \( \triangleright 3. \) Compute updated node attributes
    end for
    let \( V' = \{v'_i\}_{i=1:N^v} \)
    let \( E' = \{(e'_k, r_k, s_k)\}_{k=1:N^e} \)
    \( e' \leftarrow \rho^{e \rightarrow u}(E') \) \( \triangleright 4. \) Aggregate edge attributes globally
    \( v' \leftarrow \rho^{v \rightarrow u}(V') \) \( \triangleright 5. \) Aggregate node attributes globally
    \( u' \leftarrow \phi^u(e', v', u) \) \( \triangleright 6. \) Compute updated global attribute
    return \( (E', V', u') \)
end function
```
Practical implementation

tfgnn (TensorFlow)

jraph (Jax)

PyG (PyTorch)
