Introduction to graphs and machine learning

March 2019
Yelp, San Francisco

David Mack
david@octavian.ai

https://octavian.ai
David Mack
david@octavian.ai

https://octavian.ai
Today I’ll cover:

1. A quick intro to graphs and machine learning (ML)
2. A brief survey of current approaches to graph-ML
3. Practical challenges
Background
Knowledge

- Many ways to represent knowledge:
  - Sequences (e.g. language strings)
  - Images
  - Vectors
  - Graphs

We can contain all of these in a graph!
Knowledge graphs
Machine learning

In this presentation, we’ll mostly focus on neural networks and deep learning, a subset of machine learning.
Motivation

• Graphs are a powerful way to store data
• ML enables new forms of analysis and automation
• Let’s bring the two together!

• Also, many approaches neatly generalize when thought of in graph terms
Approaches
(This is sort of a tasting menu of approaches)

For a more thorough overview, see this article
Random walks and node embeddings
Motivating example

We want to identify clusters of people from the same neighborhood.

We know people from the same neighborhood tend to review the same restaurants.
How can we classify these subgraph nodes?
Answer: use random walks

Stumble around local neighborhood
Randomly walk the graph to linearize

Image source: https://towardsdatascience.com/node2vec-embeddings-for-graph-data-32a866340fef
Randomly walk the graph to linearize

Image source: https://towardsdatascience.com/node2vec-embeddings-for-graph-data-32a866340fef
We want to generate an embedding, i.e a vector, for each node such that nodes in the same neighborhood have similar embedding vectors:

- Assign each node $n$ an embedding $e_n$
- For each sequence $(n_1, n_2 .. n_X)$, for each pair of nodes, calculate their dot product as a score (this is the SkipGram method)
- Maximize these scores whilst minimizing the score of randomly selected (e.g. not sampled from the graph) sequences of nodes (‘negative sampling’)
- You can use gradient descent!
Embed nodes using the sequences

The most popular implementation of this is node2vec

Image source: https://towardsdatascience.com/node2vec-embeddings-for-graph-data-32a866340fef
Motivating example

We want to predict how people will rate restaurants based on past ratings
Calculate node embeddings

This is collaborative filtering using gradient descent!
Calculate embedding

Start with random embeddings for $p_i$ and $r_j$

Use gradient descent to fit $p_i$ and $r_j$ to $s_k$

Dot product $p_i$ and $r_j$ to estimate $s_k$
Motivating example

We want to recommend places to go. We know the sequences of places people have looked at/clicked in the app.
Embed nodes using naturally occurring sequences

*Image source: https://towardsdatascience.com/node2vec-embeddings-for-graph-data-32a866340fef*
Translate questions to query language
Motivating example

User asks “Who is the best barber in San Francisco”
Knowledge graph and ML responds “Andy at Joe’s Barbershop”
How clean is Spoon Street?

MATCH (var1)
WHERE var1.name="Spoon Street"
WITH 1 AS foo,
var1.cleanliness AS var2
RETURN var2

DIRTY

Translation model

Database engine
To do this

1. Generate a lot of natural language question → cypher translation pairs
2. Train your favourite sequence translation network (LSTM + attention, transformers)
3. Build guardrails for when the translation fails
Analysis of approach

Pros

• Quick to get working
• Pre-trained natural language models (e.g. BERT) can help with natural language generalization

Cons

• Textual models understanding of concepts and semantics is shallow
• System is as limited as your training data variety
Aside: Graph networks
Generalizing neural networks and graph approaches

Paper and illustrations: Relational inductive biases, deep learning, and graph networks
Embed the neural network into the graph structure

Simple example: Message passing between nodes (aka graph convolutions)

Paper and illustrations: Relational inductive biases, deep learning, and graph networks
What are graph networks able to do?

- Shortest path
- Neighborhood detection
- Node classification
- And possibly much more to be discovered

Read DeepMind’s comprehensive paper: *Relational inductive biases, deep learning, and graph networks*
Practical challenges
Practical challenges

1. No current graph databases support machine learning
2. There is no mainstream library support for graph ML
3. There’s been little (public) work on performing ML methods on production scale databases

In summary, you will need to invest a reasonable amount of work on infrastructure and research to get a production scale graph-ML system running
Q&A
Appendix
Resources

- How to get started with machine learning on graphs
CLEVR-Graph: Answering questions about mass transit graphs

- Synthetic dataset
- Question, Answer, Graph triples
- Each question comes as English, Functional program and Cypher
Seq2Seq encodes then decodes

Image source: TensorFlow tutorials
Encoding the input

Image source: TensorFlow tutorials
... then decode
In reality the output elements often derive from specific input elements.
Seq2seq Results

1. **100% translation accuracy** on (reasonably simple) CLEVR-graph question – cypher pairs
2. Google: “Human evaluations show that [Seq2Seq] has **reduced translation errors by 60%** compared to our previous phrase-based system”
Attention

1. Compare **query** to each element in array giving **scores**
2. Apply softmax to normalise and focus scores
3. Multiply each element by its score
4. Sum all the elements
Neural graph memory

- Store a table of nodes and table of edges
- Use attention (aka content addressing) to retrieve data

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>node_id</td>
<td>from_node</td>
</tr>
<tr>
<td>node_props</td>
<td>edge_props</td>
</tr>
<tr>
<td></td>
<td>to_node</td>
</tr>
</tbody>
</table>

Node table:
- node_id
- node_props

Edge table:
- from_node
- edge_props
- to_node
Let RNN cell read from a memory

The attending RNN generates a query describing what it wants to focus on.

Each item is dot produced with the query to produce a score, describing how well it matches the query. The scores are fed into a softmax to create the attention distribution.

Image source: Distil
Knowledge graphs

- Can represent a diverse range of information
- Can be continually extended
- Google’s Knowledge Graph has over 1bn entities and helps answer 30bn monthly searches
- Wikidata contains 50bn entities and is freely available