Transformers are Graph Neural Networks

Chaitanya K. Joshi

Graph Deep Learning Reading Group

Full Blogpost:
https://graphdeeplearning.github.io/post/transformers-are-gnns/
Some success stories: GNNs for RecSys
Some success stories: GNNs for RecSys
Another success story? The Transformer architecture for NLP.
Representation Learning for NLP

This \rightarrow is \rightarrow a \rightarrow sentence \quad \text{RNN}

This \rightarrow is \rightarrow another \rightarrow sentence \quad \text{Transf.}

Translation?
Sentiment?
Next word?
Part-of-speech tags?
Breaking down the Transformer

We update the hidden feature $h$ of the $i$'th word in a sentence $S$ from layer $\ell$ to layer $\ell+1$ as follows:

$$h_i^{\ell+1} = \text{Attention}(Q^\ell h_i^\ell, K^\ell h_j^\ell, V^\ell h_j^\ell),$$

i.e., $h_i^{\ell+1} = \sum_{j \in S} w_{ij} (V^\ell h_j^\ell),$

where $w_{ij} = \text{softmax}_j (Q^\ell h_i^\ell \cdot K^\ell h_j^\ell),$

where $j \in S$ denotes the set of words in the sentence and $Q, K, V$ are learnable linear weights.
Breaking down the Transformer

We update the hidden feature $h$ of the $i$'th word in a sentence $S$ from layer $\ell$ to layer $\ell+1$ as follows:

$$h_i^{\ell+1} = \text{Attention}(Q^\ell h_i^\ell, K^\ell h_j^\ell, V^\ell h_j^\ell),$$

i. e., $h_i^{\ell+1} = \sum_{j \in S} w_{ij} (V^\ell h_j^\ell),$

where $w_{ij} = \text{softmax}_j(Q^\ell h_i^\ell \cdot K^\ell h_j^\ell),$

where $j \in S$ denotes the set of words in the sentence and $Q, K, V$ are learnable linear weights.
We update the hidden feature $h$ of the $i$'th word in a sentence $S$ from layer $\ell$ to layer $\ell+1$ as follows:

$$h_i^{\ell+1} = \text{Attention}(Q^\ell h_i^\ell, K^\ell h_j^\ell, V^\ell h_j^\ell),$$

i.e.

$$h_i^{\ell+1} = \sum_{j \in S} w_{ij} (V^\ell h_j^\ell),$$

where $w_{ij} = \text{softmax}_j (Q^\ell h_i^\ell \cdot K^\ell h_j^\ell),$

where $j \in S$ denotes the set of words in the sentence and $Q, K, V$ are learnable linear weights.
Multi-head Attention

Bad random initializations can de-stabilize the learning process of this dot-product attention mechanism. We can ‘hedge our bets’ through concatenating multiple attention ‘heads’:

\[ h_{i}^{\ell+1} = \text{Concat}(\text{head}_1, \ldots, \text{head}_K)O^{\ell}, \]

\[ \text{head}_k = \text{Attention}(Q^{k,\ell}h_i^\ell, K^{k,\ell}h_j^\ell, V^{k,\ell}h_j^\ell), \]
Multi-head Attention

\[ h_{i}^{\ell+1} = \text{Concat}(\text{head}_1, \ldots, \text{head}_K) O^\ell, \]
The Final Picture

Update each word’s features through **Multi-head Attention mechanism** as a weighted sum of features of other words in the sentence.

+ Scaling dot product attention
+ Normalization layers
+ Residual links
The Final Picture

Update each word’s features through **Multi-head Attention mechanism** as a weighted sum of features of other words in the sentence.

+ Scaling dot product attention
+ Normalization layers
+ Residual links
The Final Picture

Update each word’s features through **Multi-head Attention mechanism** as a weighted sum of features of other words in the sentence.

+ Scaling dot product attention

+ Normalization layers

+ Residual links
Representation Learning on Graphs

Most influential? 😎
Least influential? 😞
Possible connection? 😊😊
Unrelated? 😞 😞
Most similar? 😊😊😊
Graph Neural Networks

GNNs update the hidden features $h$ of node $i$ at layer $\ell$ via a non-linear transformation of the node’s own features added to the aggregation of features from each neighbouring node $j \in N(i)$:

$$h_{i}^{\ell+1} = \sigma(U^{\ell} h_{i}^{\ell} + \sum_{j \in N(i)} (V^{\ell} h_{j}^{\ell})),$$

where $U$, $V$ are learnable weight matrices of the GNN layer and $\sigma$ is a non-linearity.
Connections b/w GNNs and Transformers

Consider a sentence as a fully connected graph of words...
Connections b/w GNNs and Transformers

Translation?
Sentiment?
Next word?
Part-of-speech tags?
\[ h_i^{\ell+1} = \sigma \left( U^\ell h_i^\ell + \sum_{j \in \mathcal{N}(i)} (V^\ell h_j^\ell) \right), \]

i. e., \[ h_i^{\ell+1} = \sum_{j \in S} w_{ij} (V^\ell h_j^\ell), \]

where \( w_{ij} = \text{softmax}_j (Q^\ell h_i^\ell \cdot K^\ell h_j^\ell) \),
\[ h_i^{\ell+1} = \sigma \left( U^\ell h_i^\ell - \sum_{j \in \mathcal{N}(i)} (V^\ell h_j^\ell) \right), \]

i. e., \[ h_i^{\ell+1} = \sum_{j \in S} w_{ij} (V^\ell h_j^\ell), \]

where \( w_{ij} = \text{softmax}_j(Q^\ell h_i^\ell \cdot K^\ell h_j^\ell), \)
\[ h_{i}^{\ell+1} = \sigma \left( U^{\ell} h_{i}^{\ell} + \sum_{j \in \mathcal{N}(i)} (V^{\ell} h_{j}^{\ell}) \right), \]

i. e., \[ h_{i}^{\ell+1} = \sum_{j \in S} w_{ij} (V^{\ell} h_{j}^{\ell}), \]

where \( w_{ij} = \text{softmax}_{j}(Q^{\ell} h_{i}^{\ell} \cdot K^{\ell} h_{j}^{\ell}) \),
Simple linear transform and sum

$$h_i^{\ell+1} = \sigma \left( U_\ell h_i^\ell + \sum_{j \in N(i)} (V_\ell h_j^\ell) \right),$$

Weighted sum via attention

i. e., $h_i^{\ell+1} = \sum_{j \in S} w_{ij} (V_\ell h_j^\ell),$

where $w_{ij} = \text{softmax}_j (Q_\ell h_i^\ell \cdot K_\ell h_j^\ell),$
\[ h^{\ell+1}_i = \sigma \left( U_\ell h^\ell_i + \sum_{j \in \mathcal{N}(i)} \left( V_\ell h^\ell_j \right) \right), \]

\[ i.e., h^{\ell+1}_i = \sum_{j \in S} w_{ij} \left( V_\ell h^\ell_j \right), \]

where \( w_{ij} = \text{softmax}_j(Q_\ell h^\ell_i \cdot K_\ell h^\ell_j) \),
Standard GNN

- Weighted sum aggregation

GAT

- Multi-head mechanism
- Normalization layers
- Residual links

Transformer

- LayerNorm
- FF-MLP
- Concat$_K$
- Scaled Dot Product
- softmax$_j$

- $h_{i+1}^f$
- $h_{i+1}^f$
- $K^n$
- $K^n$
- $V^n$
- $V^n$
- $V^n$
- $V^n$