Graph Learning Machine

Homework number 4

Writer:

Mohammad Rezaei Kalantary

Student Number:

401422087

Teacher:

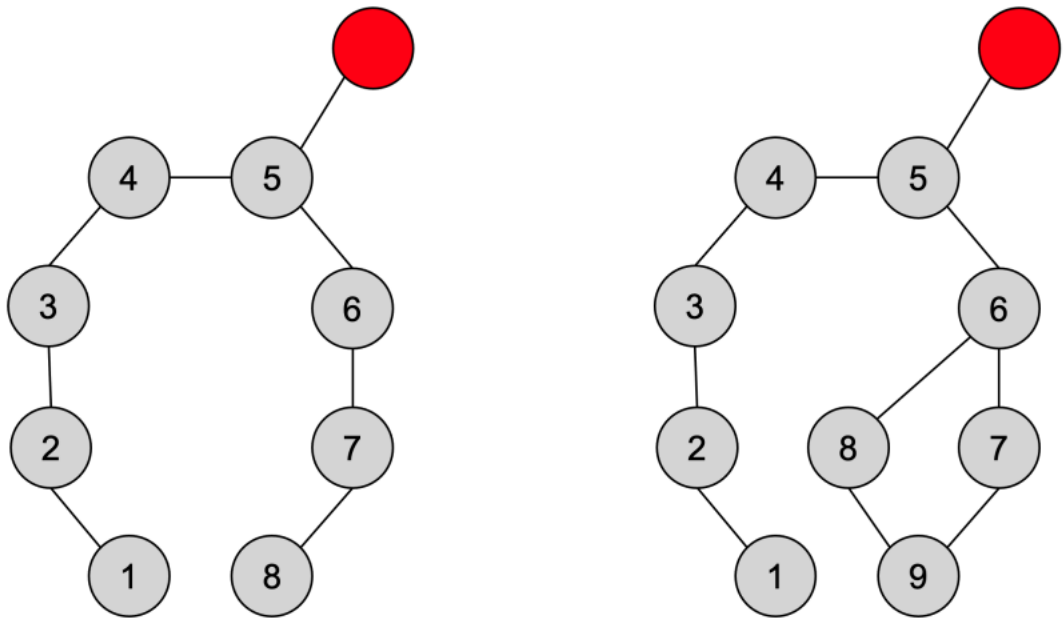
Dr Zahra Taheri

Shahid Beheshti University
Faculty of Mathematics

Spring 1401

Effect of Depth on Expressiveness

Consider the following 2 graphs, where all nodes have 1-dimensional initial feature vector $x = [1]$. We use a simplified version of GNN, with no nonlinearity, no learned linear transformation, and sum aggregation. Specifically, at every layer, the embedding of node v is updated as the sum over the embeddings of its neighbors (N_v) and its current embedding h_v^t to get h_v^{t+1}. We run the GNN to compute node embeddings for the 2 red nodes respectively. Note that the 2 red nodes have different 5-hop neighborhood structure (note this is not the minimum number of hops for which the neighborhood structure of the 2 nodes differs). How many layers of message passing are needed so that these 2 nodes can be distinguished (i.e., have different GNN embeddings)?



Answer : If we draw the GNN embeddings from the starting red point we will reach the distinguished between 2 graph in layer 3

Relation to Random Walk

Let $h^{(1)_i}$be the embedding of node e i at layer l. Suppose that we are using a mean aggregator for message passing, and omit the learned linear transformation and non-linearity:$h^{(l+1)_i} = \frac{1}{|N_i| \sum_{j \in N_i} h_j^l}$. If we start at a node u and take a uniform random walk for 1 step, the expectation over the layer-l embeddings of nodes we can end up with is$h^{(1+1)_u}$, exactly the embedding of u in the next GNN layer. What is the transition matrix of the random walk? Describe the transition matrix using the adjacency matrix A, and degree matrix D, a diagonal matrix where Di,i is the degree of node i.

Answer : $D^{-1}A$

Learning BFS with GNN

Consider breadth-first search (BFS), where at every step, nodes that are connected to already visited nodes become visited. Suppose that we use GNN to learn to execute the BFS algorithm. Suppose that the embeddings are 1-dimensional. Initially, all nodes have input feature 0, except a source node which has input feature 1. At every step, nodes reached by BFS have embedding 1, and nodes not reached by BFS have embedding 0. Describe a message function, an aggregation function, and an update rule for the GNN such that it learns the task perfectly.

Answer :
Message Function :

$$m_u^{(l)} = MSG^{(l)}(h_u^{l-1})$$

Aggregation Function :

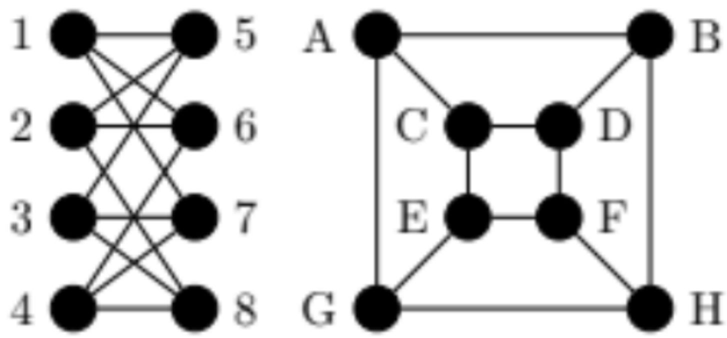
$$AGGREGATE_{max}((h_u^{l-1}, \forall u \in N(v)))_i = max_{u \in N(v)}(h_u^{l-1})$$

Update Rule :

$$h_v^{(l)} = Max(m_u^l, u \in N(v))$$

Isomorphism Check

Are the following two graphs isomorphic? If so, demonstrate an isomorphism between the sets of vertices. To demonstrate an isomorphism between two graphs, you need to find a 1-to-1 correspondence between their nodes and edges. If these two graphs are not isomorphic, prove it by finding a structure (node and/or edge) in one graph which is not present in the other.



Answer : One way to answer this question is assigning each node of the first graph to each node of second graph and check how does the edges follows. For this special case we Consider

$$A \to 1$$

$$B \to 5$$

$$C \to 6$$

$$G \to 7$$

$$D \to 3$$

$$H \to 2$$

$$E \to 4$$

$$F \to 8$$

And this will shows that this two graphs are indeed Isomorph.

Aggregation Choice

The choice of the AGGREGATE(·) is important for the expressiveness of the model. Give an example of two graphs $G_1 = (V_1, e_1)$ and $G_2 = (V_2, E_2)$ and their initial node features, such that for two nodes $v_1 \in V_1$ and $v_2 \in V_2$ with the same initial features $h_{v1}^{(0)} = h_{v2}^{(0)}$, the updated features $h_{v1}^{(1)}$ and $h_{v1}^{(1)}$ are equal if we use mean and max aggregation, but different if we use sum aggregation.

Answer : In the following Graphs we see that in figure 1 We have 5 node that for of them are connected to one middle red node so consider the Aggregate methods We will have $AGG_{Max} = 2$ and $AGG_{Mean} = 2$ and $AGG_{Sum} = 6$ but in Figure 2 these numbers will be 3 node that for of them are connected to one middle red node so consider the Aggregate methods We will have $AGG_{Max} = 2$ and $AGG_{Mean} = 2$ and $AGG_{Sum} = 4$

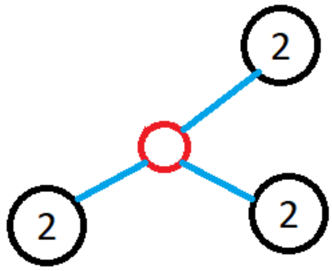


Figure 1: "5 Node"

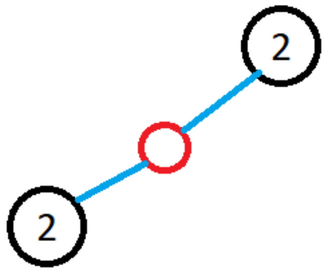


Figure 2: "3 Node"

Weisfeiler-Lehman Test

Our isomorphism-test algorithm is known to be at most as powerful as the well known Weisfeiler-Lehman test (WL test). At each iteration, this algorithm updates the representation of each node to be the set containing its previous representation and the previous representations of all its neighbours.Prove that our neural model is at most as powerful as the WL test. More precisely, let G1 and G2 be non-isomorphic, and suppose that their node embeddings are updated over K iterations with the same AGGREGATE(·) and COMBINE(·) functions. Show that if

$$READOUT(h_v^{(k)}, \forall v \in V_1) \neq READOUT(h_v^{(k)}, \forall v \in V_2)$$

then the WL test also decides the graphs are not isomorphic.

Answer : We will use proof by contradiction by first assuming that Weisfeiler-Lehman test cannot decide whether G1 and G2 are isomorphic at the end of K'th iteration. Let's say Weisfeiler-Lehman test is unable to decide whether G1 and G2 are isomorphic at the end of K'th iteration. Considering the result of the algorithm we will have 2 embbeded Graph with atleast 2 feature representation vecotrs at the end of K'th iteration. Then would We have 2 different results.

result n1: If the 2 feature representation vecotrs were identical, this would mean that for each node of G1 and edges connected this it we will have same node in G2 that carry same feature representation. So in conclusion the test can decide if they are Isomorph.

result n2:If the 2 feature representation vecotrs were'nt identical, this would mean that there would be atleast one node in G1 or G2 that doesn't carry same feature representation. There goes that the test can decide if they are not Isomorph.

Now according to the details in our question we have 2 graphs with same iterations of embeddings and same AGGREGATE and COMBINE functions are not isomorph And according the proof in our WL-test it also can decide if they are Isomorph or not.