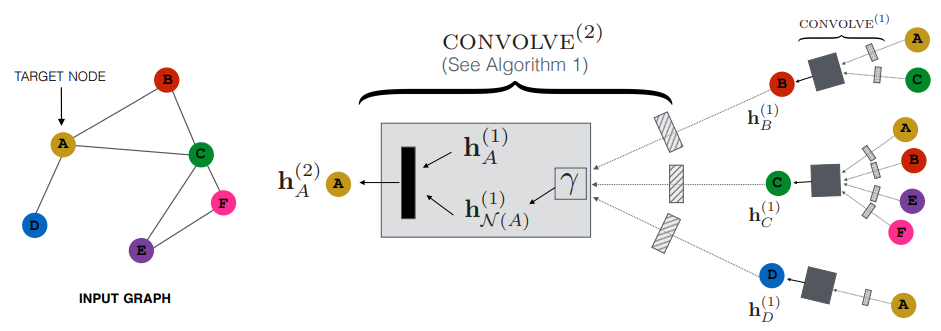
John (Ziyuan) Zhou

Week 9 Summary

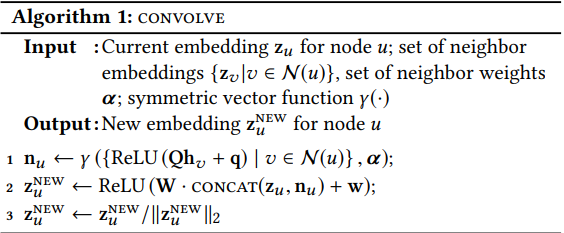
UW ID: 2150115

In the article “Graph Convolutional Neural Networks for Web-Scale Recommender Systems”, the authors point out the problem that existing method is difficult to be practical and scalable to web-scale recommendation tasks with large numbers of items and users, although they achieved state-of-the-art performance on recommender system benchmarks. Therefore, they propose a highly-scalable GCN framework named PinSage, which operates on a massive graph with 3 billion nodes and 18 billion edges—a graph that is 10, 000× larger than typical applications of GCNs. The core idea is combining efficient random walks and graph convolutions to generate embeddings of nodes that incorporate both graph structure as well as node feature information.

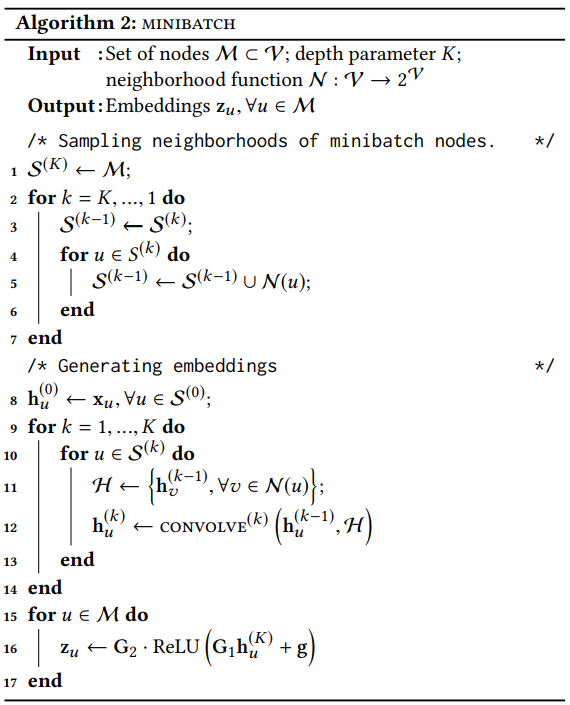
For the model architecture, the authors use localized convolutional modules to generate embeddings for nodes. They start with input node features and then learn neural networks that transform and aggregate features over the graph to compute the node embeddings.



* Forward propagation algorithm: The core of the PinSage algorithm is a localized convolution operation, where learns how to aggregate information from node’s neighborhood. The output of the algorithm is a representation of that incorporates both information about itself and its local graph neighborhood.



* Importance-based neighborhoods: In PinSage, the authors define importance-based neighborhoods, where the neighborhood of a node is defined as the nodes that exert the most influence on node . Concretely, they simulate random walks starting from node u and compute the L1-normalized visit count of nodes visited by the random walk. The neighborhood of is then defined as the top nodes with the highest normalized visit counts with respect to node .
* Stacking convolutions: Each time we apply the convolve operation, we get a new representation for a node, and we can stack multiple such convolutions on top of each other to gain more information about the local graph structure around node . The following algorithm shows how stacked convolutions generate embeddings for a minibatch set of nodes.



The loss function for each pair of node embeddings is the same as follow:

* For a single pair of node embedding , .

For the experiment and results, in offline ranking metrics, PinSage improves over the best performing baseline by more than 40%. And in head-to-head human evaluations, authors’ recommendations are preferred about 60% of the time. The A/B tests show 30% to 100% improvements in user engagement across various settings.