Arnaud Massenet
arnaudyoh@gmail.com

Lab session # 6
ALTEGRAD 2019

29/12/19

Question 1

In order to avoid situations where $v_{i+2} = v_i$, we can ignore the previously visited node from the list neighbors when choosing the next node to visit. By doing so we prevent ourselves to visit the same node again and again. Note however that we cannot do this if a node has only one neighbor and said neighbor has already been visited.

Question 2

In [3], the authors propose an architecture for classifying graphs that is very similar to the one we use for the node classifcation. To classify graphs, the authors use a SortPooling layer followed by a 1D convolution and a dense layer. As such, in our model, we can switch our third layer with a SortPooling layer followed by a 1D convolution and a dense layer. Additionally, instead of simply using the output of the last graph convolution we will concatenate the outputs of our 2 messages passing layers and pass them to the SortPooling layer.
As a side note the SortPooling layer functions as follows: we sort each of the feature descriptors using their Weisfeiler and Lehman "colors" [1]. These "colors" define an ordering based on the graph topology. As such the SortPooling layer will sort nodes based on their structural roles in the graph.

Question 3

The GNN did outperform the DeepWalk + logistic regression in the node classification task as it achieves 1.0 accuracy on the test set compared to .81 with the DeepWalk approach. This is mainly because the architecture of our neural network is better able to capture the important features of each nodes since the neural network allows more complex representation of the data than a simple logistic regression. In [2], we have that the GNN under certain condition can be as powerful as the WL test for distinguishing graphs. As such, we expect that the GNN can be very good at discerning the structural differences between nodes and classify them accordingly.

Question 4

When we defined the features as a diagonal matrix we achieved 1.0 accuracy on the test set. With the features matrix defined with all ones, we only achieve 0.286 accuracy. When setting all feature vectors to be the same, we are essentially saying that all nodes in the graph are the same. By making all nodes the same, the model will have trouble distinguishing the structural role of each node within the graph, which explains the poor performance.

Figures

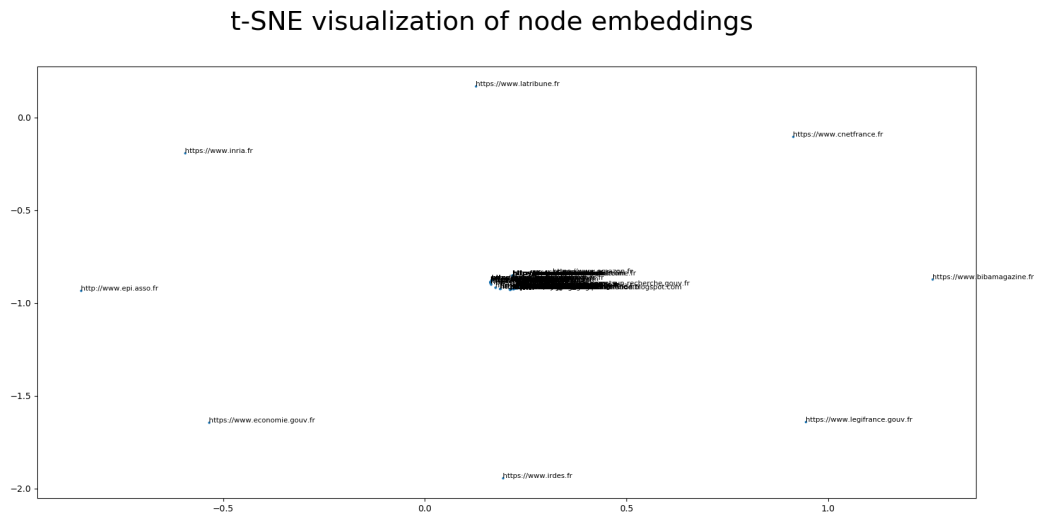

Figure 1: French Web Visualization

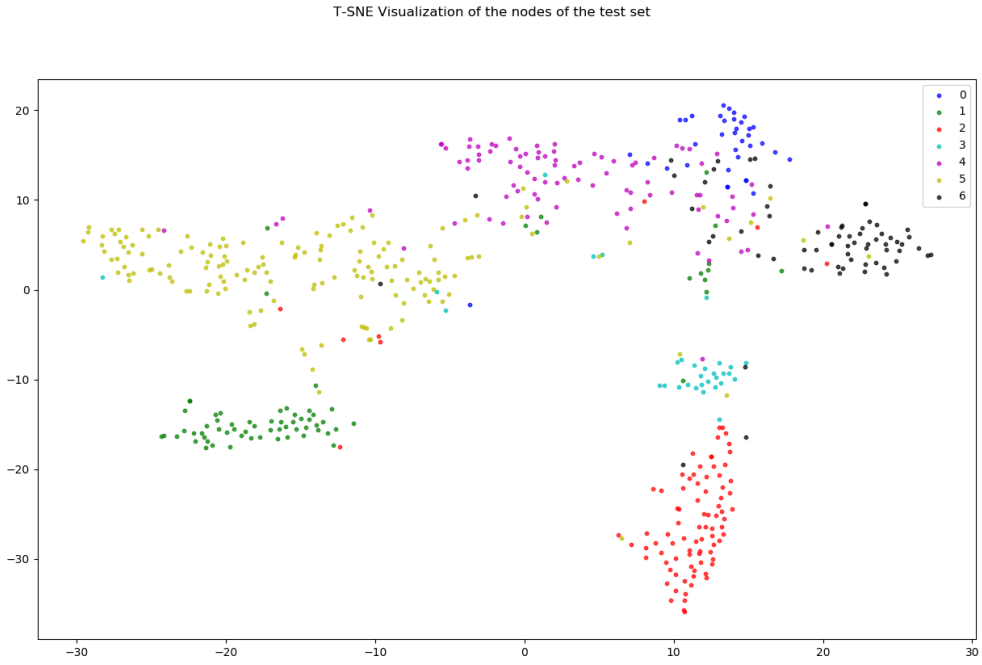

Figure 2: Cora t-SNE

References

[1] B. Yu. Weisfeiler and A. A. Leman. Reduction of a graph to a canonical form and an algebra arising during this reduction. 1968.

[2] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *CoRR*, abs/1810.00826, 2018.

[3] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification, 2018.