Laurine Marty
Entry n° 2022VST9547

COL870 – Sp. Topic in Machine Learning
Model Bench-marking :

SpeqNets

The purpose of this report is to study the performance of the SpeqNets models, introduced in the paper "SpeqNets : Sparsity-aware Permutation-equivariant Graph Networks", published in 2022 by Christopher Morris et. al.
It will be compared with the GCN model on node classification and graph similarity learning tasks.

1. SpeqNets models

Motivated by the idea that massage-passing neural networks have clear limitations in approximating permutation-equivariant function over graphs and that more expressive, high-order graph neural networks do not scale to large graphs, the SpeqNets were thinked to be a fine-grained solution between expressivity and scalability and adapt to the sparsity of the graph.
The neural architecture that can reach the level of the k-WL in terms of separating the non-isomorphic graphs have a memory reuierement lower-bounded by n^k, for a n-node graph. Moreover, the cardinality of the local neighborhood is always k*n.
In 2020, Morris et al. [1] introduced the local variant (δ-k-LWL) of the k-WL considering only a subset of the neighborhoods in k-WL. However, like the original algorithm, the local variant operates on the set of all possible k-tuples, again resulting in the same (exponential) memory requirements, rendering the algorithm not practical for large, real-world graphs.

Thus, to adress the memory problem, the SpeqNet model is presented as a new set of heuristics for the graph isomorphic problem, denoted (k,s)-LWL, which only considers a subset of all k-tuples, namely those inducing subgraphs with at most s connected component. The hyper-parameters k and s could permit to fit the sparsity of each graphs. Subsequently the paper presents the corresponding provably expressive, permutation-equivariant neural architecture : (k,s) – SpeqNets.

The SpeqNet architecture computes representation for k-tuples. However it is possible to derive neural architectures based on the (k-s)-LWL for the nodes and edges level learning tasks. Given a graph G, to learn a node feature for node v, we can simply pool over the feature learned for (k, s) tuples containing the node v as a component. That is, let t ≥ 0, then we consider the multisets for I in [k] :

$$m^t(v)_i = \{\!\{ f^{(t-1)}(\mathbf{t}) \mid \mathbf{t} \in V(G)^k_s \text{ and } t_i = v \}\!\}$$

$V(G)^k_s$: is the set of (k, s)-tuples of nodes, i.e, k-tuples which induce (sub-)graphs with at most s (connected) components.

Hence, to compute a vectorial representation of the node v, the model compute a vectorial representation of the related multiset, using a neural architecture for multiset presented in a paper published in 2020 by Azizan and Lelarge [2], followed by learning a join vectorial representation for the node v.

2. GCN model

As a baseline, we chose to use the GCN architecture, presented in the paper published in 2016 by Thomas Kipf et. al. [3]. For the experimentation we implemented a model compounded by two GCN convolutional layer and the ReLu activation function.

3. Problem

Considering the characteristics of the SpeqNet model, we aim to study the following points.

- The impact on the performances of the use of a given SpeqNet model (k and s fixed) on the given tasks : Node classification and graph similarity learning.
- Its impact on the computational time
- The impact of the k and s parameters on the computational time and the performances.

4. Task 1 : Nodes classification

To benchmak SpeqNet model on the node classification task we use two datasets :

- Cora : A citation network dataset
- WebKB – Wisconsin : a dataset where nodes represent webpages and edges are the hyperlinks between them.

	Nodes	Edges	Features dimension	Number of classes
Cora	2708	10 556	1 433	7
WebKB – Wisconsin	251	515	1 703	5

For more information about the loss function and the metric used, one can refer to the assignment 2 statement.

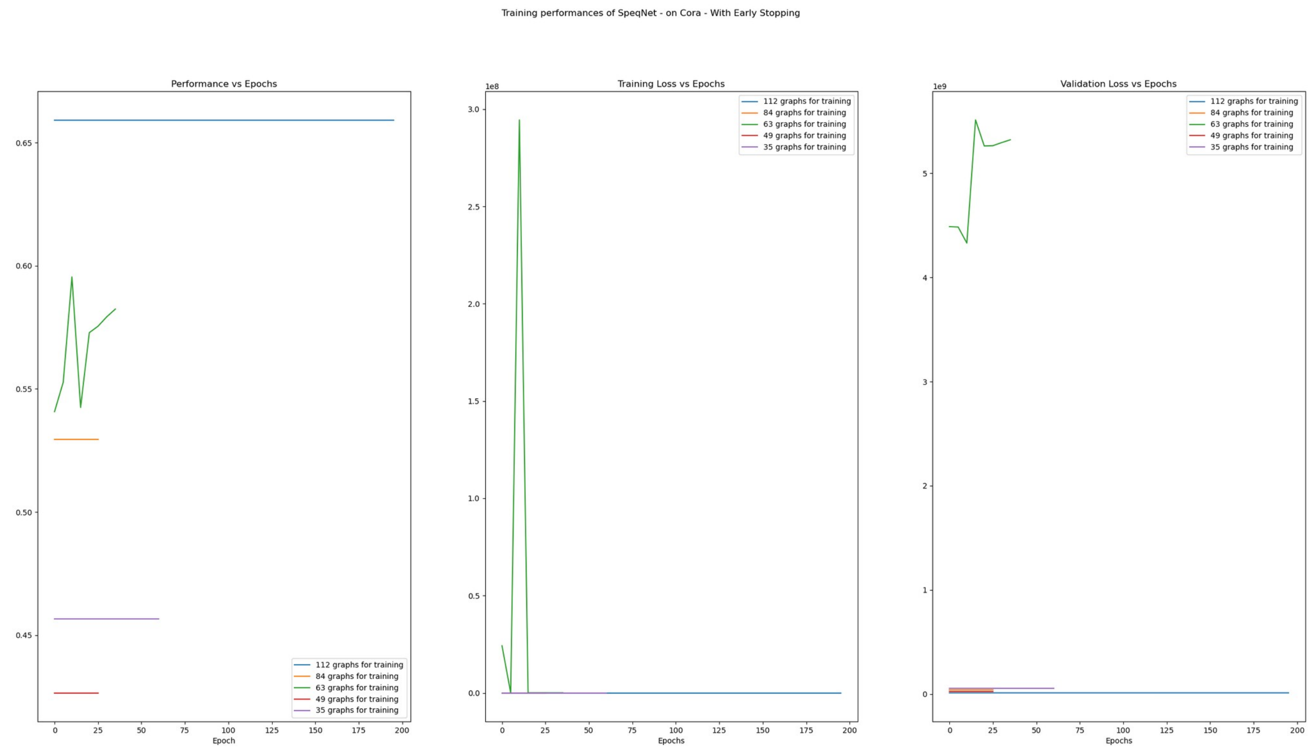
For this task, we used at first the (2,1)-SpeqNet model, less demanding in terms of memory.

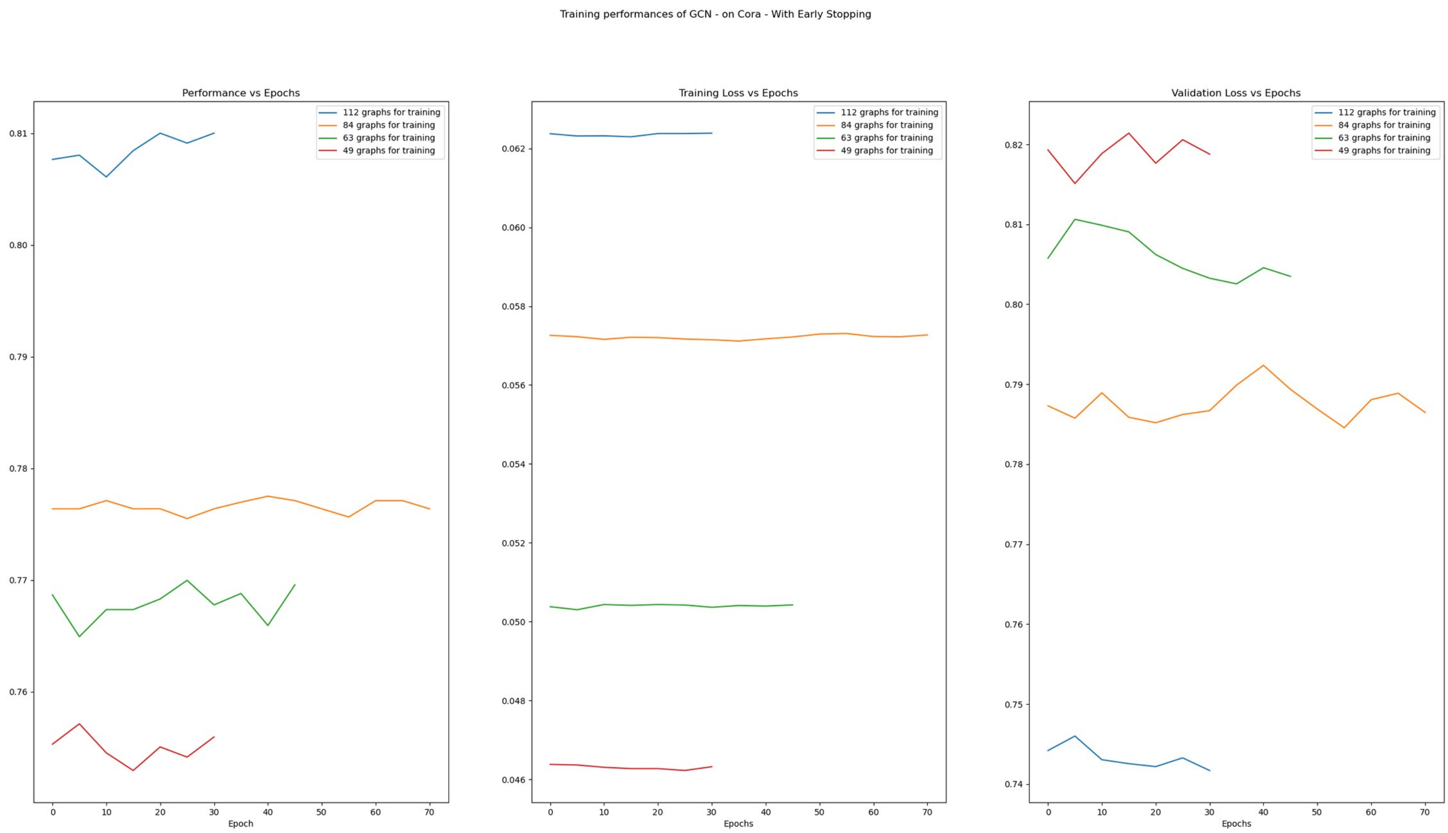
In the following part, the results of the experience are presented.

Performances of the two models on the node classification task :

	Cora		WebKB – Wisconsin	
	Accuracy (%)	F1 – Score	Accuracy (%)	F1 – Score
GCN	81,09	0,80	95,16	0,80
(2,1)-SpeqNet	65,63	0,61	100	0,96

Plot of the evolution of the performances according to the volume of training data :





5. Task 2 : Graph Similarity Learning

To benchmark the SpeqNet model on the Graph Similarity Learning we use two datasets :

- LINUX
- AIDS700nef

	Graphs	Nodes	Edges	Features	Classes
LINUX	1000	~7,6	~13,9	0	0
AIDS700nef	700	~8,9	~17,6	29	0

For the LINUX dataset, the initialization of the feature has been made by computing the vector that gave the degree of each nodes in the graph, followed by a padding.

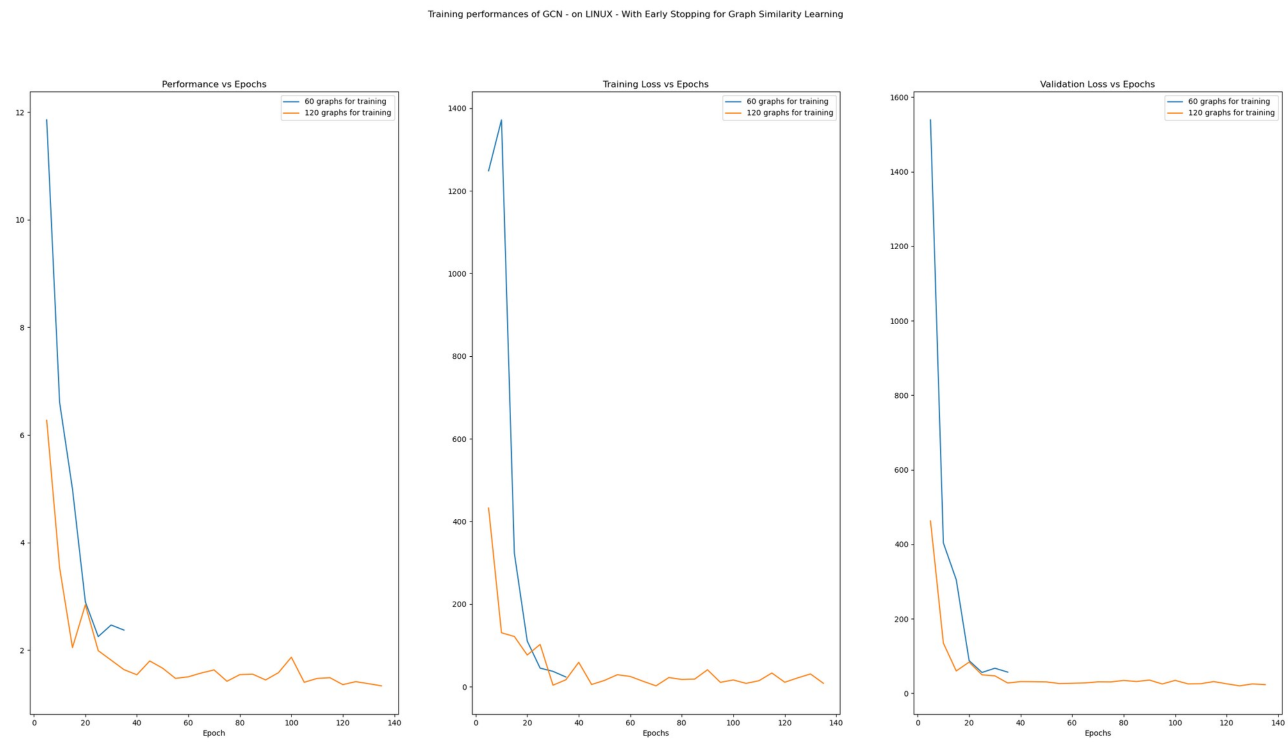
We also used the (2,1)- SpeqNet model during this experience.

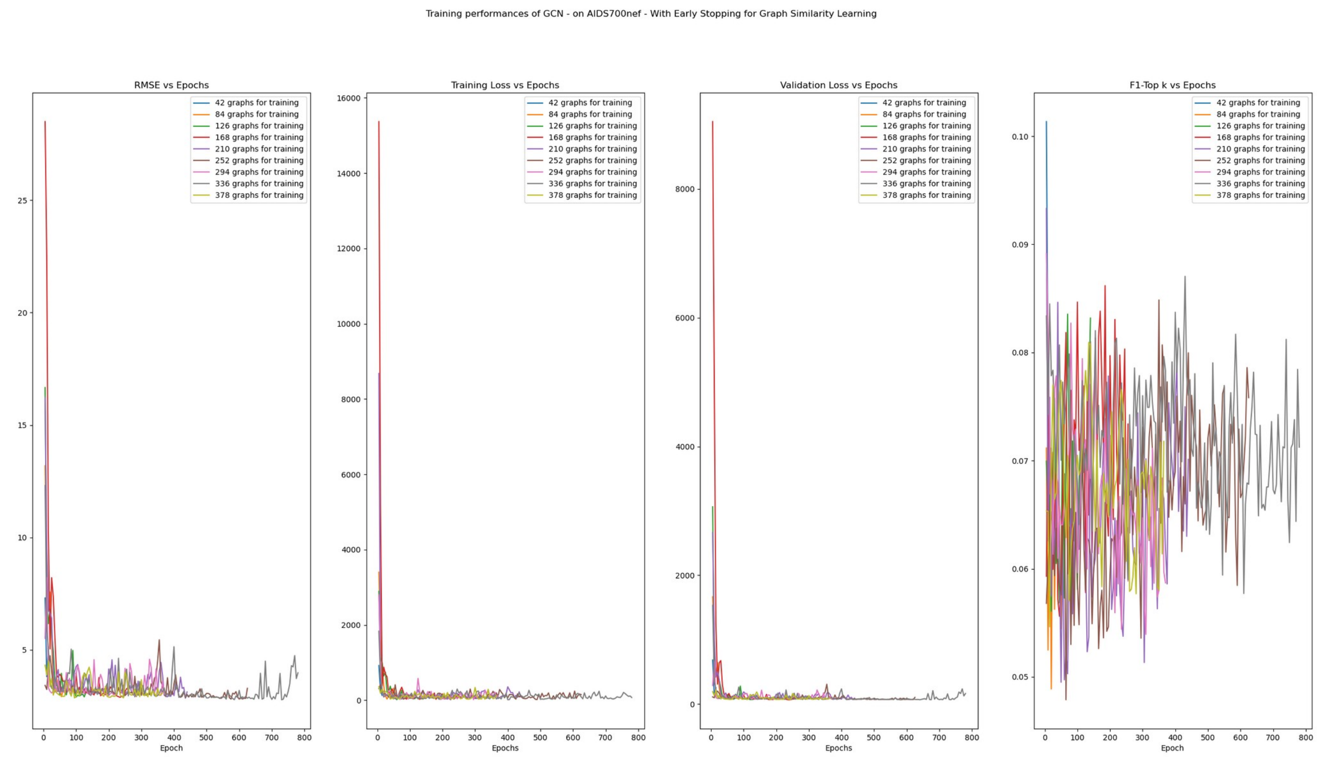
For more information about the loss function and the metric used, one can refer to the assignment 2 statement.

In the following part, the results of the experience are presented.

	LINUX	AIDS700nef
GCN	RMSE : 1,31	RMSE : 2,5
(2,1)-SpeqNet	Training Time > 2 days	

Plot of the evolution of the performances according to the volume of training data :




Training performances of GCN - on AIDS700nef - With Early Stopping for Graph Similarity Learning
RMSE vs Epochs
Epoch
Training Loss vs Epochs
Epochs
Validation Loss vs Epochs
Epochs
F1-Top k vs Epochs
Epoch
42 graphs for training
84 graphs for training
126 graphs for training
168 graphs for training
210 graphs for training
252 graphs for training
294 graphs for training
336 graphs for training
378 graphs for training

[1] C. Morris, G. Rattan, and P. Mutzel. Weisfeiler and Leman go sparse: Towards higher-order graph embeddings. In Advances in Neural Information Processing Systems, 2020.

[2] W. Azizian and M. Lelarge. Characterizing the expressive power of invariant and equivariant graph neural networks. CoRR, abs/2006.15646, 2020.

[3] Thomas N. Kipf, Max Welling , Semi-Supervised Classification with Graph Convolutional Networks, 2016