Catastrophic Forgetting in Deep Graph Networks: an Introductory Benchmark for Graph Classification


Antonio Carta


Andrea Cossu


Federico Errica

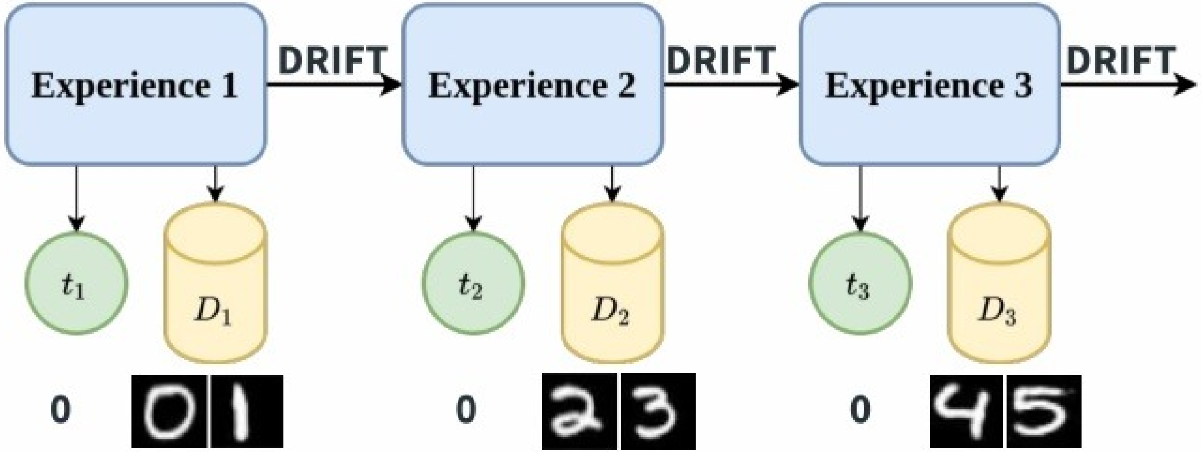

Davide Bacciu

***Graph Learning Benchmarks** Workshop, **WWW 2021**.*

Outline

1. CONTINUAL LEARNING BACKGROUND
2. PAPER'S AIMS
3. EXPERIMENTAL SETUP
4. RESULTS
5. FUTURE DIRECTIONS
6. Q&A

Continual Learning setting



WE STUDIED CATASTROPHIC FORGETTING

Why bothering?

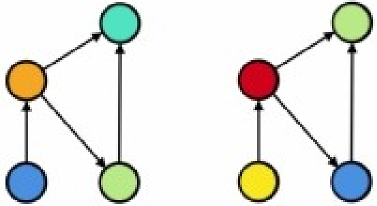
Training on the entire data may even lead to superior results! However…

- Datasets may be **huge** → **no training on the edge** 😔
- **New data** after deployment → **retraining**
    - **Expensive** 😔
    - **Inefficient** → most of the **information** is **already in** the **model** 😔

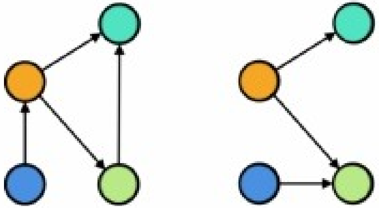
Paper's Aim (1)

Does CL work on graphs?

- **Graph** distribution drift
- GRL **regularization** strategies?
- Graph **classification**



node distribution drift



neighborhood distribution drift

Paper's Aim (2)

Reproducible Research + Framework

- **Foster** future research
- Avoid **common mistakes**
- Handle boilerplate code

PYDGN 💙 CL

https://github.com/diningphil/continual_learning_for_graphs

CL + GRL techniques

Elastic Weight Consolidation (EWC) → $$\mathcal{R}(\Theta, \Omega) = \lambda \sum_{i=1}^{n-1} \Omega_\mathbf{i} \|\Theta_\mathbf{i} - \Theta_\mathbf{n}\|_2^2$$

Learning without Forgetting (LwF) → $$\mathcal{R}(\Theta_\mathbf{n}, \Theta_\mathbf{n-1}; \mathbf{x}, \mathbf{y}) = \alpha \,\mathrm{KL}[p_{\Theta_\mathbf{n}}(\mathbf{y}|\mathbf{x}) \,||\, p_{\Theta_\mathbf{n-1}}(\mathbf{y}|\mathbf{x})]$$

Structure Preserving Regularization (REG) → $$p\,(\mathbf{A} \,|\, \mathbf{Z}) = \prod_{i=1}^{N} \prod_{j=1}^{N} p\,(A_{ij} \,|\, \mathbf{z}_i, \mathbf{z}_j)$$

$$\text{with} \quad p\,(A_{ij} = 1 \,|\, \mathbf{z}_i, \mathbf{z}_j) = \sigma(\mathbf{z}_i^\top \mathbf{z}_j)$$

[Ref. 3]

Replay Memory **Naive Strategy**

Which Models?

Structure-agnostic Baseline

- Impact of structure in **CL**
- MLP + Mean Global Pooling

$$\mathbf{h}_v = \psi(\mathbf{x}_v), \quad x_v \in \mathcal{X}_g,$$

$$\psi(x_v) = \mathbf{W}_L^T(\sigma(\ldots(\sigma(\mathbf{W}_1^T x_v + \mathbf{b}_1)\ldots) + \mathbf{b}_L)$$

$$\mathbf{h}_g = \Psi_g\Big(\{\mathbf{h}_v \mid v \in \mathcal{V}_g\}\Big)$$

Generic and simple DGN

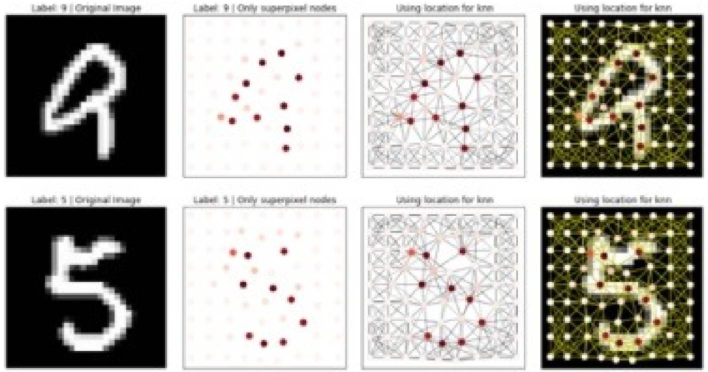
- Based on GraphSAGE
- Mean aggregator & Global Pooling

$$\mathbf{h}_v^{\ell+1} = \phi^{\ell+1}\Big(\mathbf{h}_v^{\ell}, \ \Psi_n(\{\psi^{\ell+1}(\mathbf{h}_u^{\ell}) \mid u \in \mathcal{N}_v\})\Big)$$

$$\mathbf{h}_g = \Psi_g\Big(\{\mathbf{h}_v \mid v \in \mathcal{V}_g\}\Big)$$

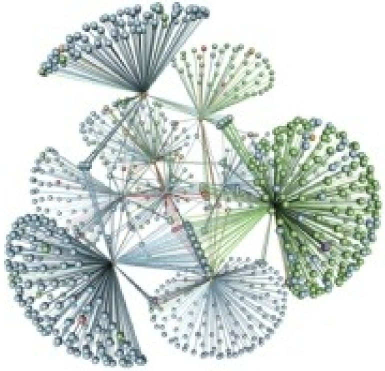
MLP
GRAPH CONVOLUTION(S)
GLOBAL POOLING

Datasets



(a) MNIST [Ref. 5]

CIFAR10 (same preprocessing of MNIST)



OGBG-PPA

Evaluation Setup

- Class-incremental Scenario
- Monitor $\text{ACC} = \frac{1}{T}\sum_{t=1}^{T} R_{T,t}$
- Hold-out + Hyper-param optimization **for all models**

	MNIST	CIFAR10	OGBG-PPA
Size	70000	60000	158100
Node Attrs.	3	5	0
Edge Attrs.	0	0	7
Classes	10	10	37
Avg $\lvert\mathcal{V}_g\rvert$	70,57	117,63	243,4
Avg $\lvert\mathcal{E}_g\rvert$	564,63	941,07	2266,1
Data Split	55K/5K/15K	45K/5K/15K	49%/29%/22%
Class Split	2+2+2+2+2	2+2+2+2+2	17+5+5+5+5

Results

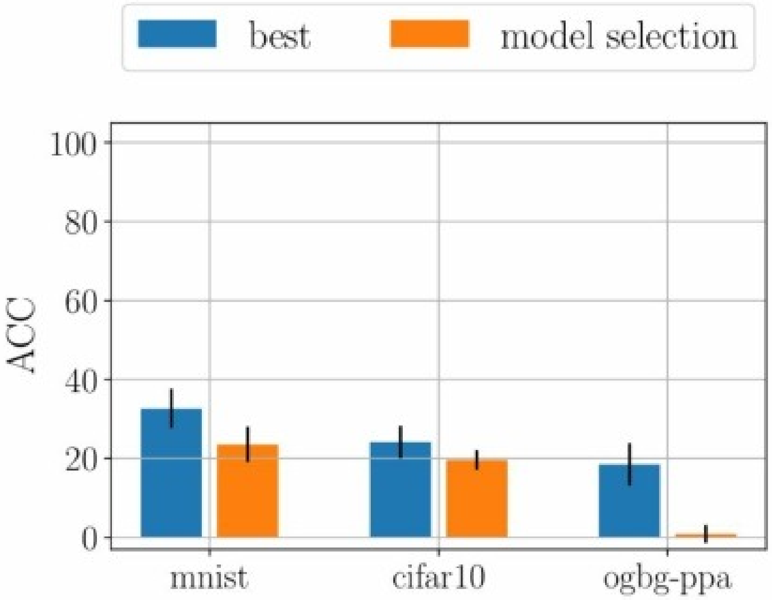
Baseline: Competitive!

Are DGNs ignoring the structure?

	Model	Naïve	EWC	Replay	LWF
MNIST	Baseline	$19.56_{\pm 0.1}$	$19.39_{\pm 0.1}$	$86.13_{\pm 4.5}$	$33.16_{\pm 13.1}$
	DGN	$19.19_{\pm 0.1}$	$18.95_{\pm 0.3}$	$79.52_{\pm 1.9}$	$32.64_{\pm 5.0}$
	DGN+reg	$19.31_{\pm 0.1}$	–	$81.42_{\pm 2.4}$	–
CIFAR10	Baseline	$17.49_{\pm 0.1}$	$17.49_{\pm 0.1}$	$42.87_{\pm 3.7}$	$26.77_{\pm 5.1}$
	DGN	$17.11_{\pm 0.2}$	$17.10_{\pm 0.2}$	$39.55_{\pm 2.3}$	$24.13_{\pm 4.1}$
	DGN+reg	$17.13_{\pm 0.1}$	–	$46.61_{\pm 3.5}$	–
OGBG-PPA	Baseline	$14.53_{\pm 0.5}$	$13.90_{\pm 0.8}$	$55.96_{\pm 3.0}$	$20.83_{\pm 6.1}$
	DGN	$14.47_{\pm 0.3}$	$14.15_{\pm 0.5}$	$56.34_{\pm 2.5}$	$18.46_{\pm 5.4}$
	DGN+reg	$15.18_{\pm 0.8}$	–	$57.27_{\pm 3.2}$	–

Table 2: Mean accuracy and mean standard deviation (in parenthesis) among all steps. Replay results are related to memory size of 1000. Results are averaged over 5 final runs. We treat the regularization loss as a separate strategy.

Results: LwF sensitivity to hyper-parameters



The road ahead

- Better understand the **role of graph distribution drift** on forgetting
  - Design ad-hoc regularization
- **Need more benchmarks!**
- Study **node classification** and other tasks

References

1. Cossu A., Carta A., Errica F., Bacciu D., "*Catastrophic Forgetting in Deep Graph Networks: an Introductory Benchmark for Graph Classification*", ***Graph Learning Benchmarks*** *Workshop*, ***WWW*** *2021.*

2. Bacciu D., Errica F., Micheli A., Podda M., *"A Gentle Introduction to Deep Learning for Graphs"*, ***Neural Networks***, *2021*

3. Kipf, T. N., Welling, M., *"Variational Graph Auto-Encoders"*, ***Bayesian Deep Learning*** *Workshop*, ***NIPS***, *2016*

4. Hamilton, W., Ying, Z., Leskovec, J., *"Inductive representation learning on large graphs"*, ***NIPS***, *2017*

5. Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y., & Bresson, X., *"Benchmarking Graph Neural Networks", arXiv 2020.*

Questions?

Thank you!

You can reach me at:

andrea.cossu@sns.it

andreacossu.github.io

CIML homepage: ciml.di.unipi.it

PAI Lab: http://pai.di.unipi.it/