Sapienza Università di Roma

Advanced Machine Learning

Graph MetaNetworks for unlearning

Group Members:
Laura Thoft Cesario – 1852596
Diego Ciciani – 2140394
Livia Oddi - 1846085
Damian Zeller - 2118831

Professor:
Galasso Fabio

Tutors:
Alessio Palma
Leonardo Plini

November 22, 2024

Contents

1 General overview

This project explores **unlearning mechanisms using Graph Meta Networks (GMNs)**, focusing on the **Cora citation dataset**. GMNs enable structured representation of data as graphs, where nodes and edges encode complex relationships, facilitating selective unlearning of specific classes while retaining performance on other data [1].

In this study, we target the unlearning of the **1st class (Neural Networks)** in the Cora dataset, while evaluating the impact on the remaining classes. Additionally, the project **integrated the pre-trained graph meta network (GMN) with the Cora dataset** to enhance the model's understanding of graph structures and relationships. The **vector** u was used to one-hot encode the class to be forgotten, therefore serving as a global indicator to guide the model on which class to forget during the unlearning process.

2 Dataset description

The **Cora dataset** is a citation network that represents scientific papers categorized into multiple research topics. Its graph-based structure makes it particularly suitable for studying graph models, as the nodes represent documents (scientific papers) and the edges encode relationships (citations) between these papers. This inherent connectivity allows for the exploration of complex interactions and dependencies within the dataset, making it ideal for tasks like *unlearning*.

Each **node** in the dataset is assigned a class label corresponding to one of seven research topics, such as Neural Networks, Rule Learning, or Probabilistic Methods. The **edges**, on the other hand, represent citation relationships, forming an undirected connection between two papers if one cites the other. This network structure provides valuable insights into how research papers are interconnected based on their references.

The **features of each node** are represented as a bag-of-words vector, reflecting the presence of specific words in the corresponding paper. The dataset's **feature matrix** has dimensions of (2708, 1433), where *2708* corresponds to the *number of nodes (papers)* and *1433* represents the *number of unique words used across all documents*. These features encapsulate the textual content of the papers, enabling meaningful analysis and learning.

The dataset includes **7 classes**, indexed from 0 to 6, each representing a specific research topic. However, the distribution of these classes is imbalanced, with certain topics being overrepresented compared to others.

2.1 Summary of dataset statistics

Statistic	**Value**
Number of Nodes	2708
Number of Edges	10556
Number of Node Features	1433
Number of Classes	7
Edge Type	Undirected

Table 1: Key characteristics of the Cora dataset

The Cora dataset offers a structured and well-defined environment for experimenting with graph-based models. Its citation network structure provides a natural graph representation of data, enabling the application of graph neural networks (GNNs). The diversity of node features and the well-defined class labels make it particularly suitable for tasks like node classification and selective unlearning. Moreover, the imbalanced class distribution adds complexity, reflecting real-world scenarios where certain categories of data may dominate others.

Using the Cora dataset allows us to focus on the challenge of selectively forgetting specific classes while retaining performance on the rest. Its compact size and graph structure make it computationally efficient to train and evaluate graph models, providing an excellent platform for studying unlearning mechanisms [2].

3 Methodology and design

3.1 Graph representation

The Cora dataset is represented as a graph where nodes correspond to documents, and edges represent citation relationships between these documents. Each node is enriched with features derived from the bag-of-words representation of the corresponding paper. These features capture the textual content, providing a meaningful input for graph-based learning models.

The Cora dataset was chosen for this project because of its inherent graph structure, which makes it particularly well-suited for graph-based machine learning models. Additionally, the dataset provides a clear and intuitive use case for unlearning due to its well-defined class labels, which represent distinct research topics. This structure allows for targeted forgetting, making it an ideal benchmark for evaluating unlearning algorithms.

3.2 Graph Meta Network

The Graph Meta Network (GMN) architecture used in this project combines two essential components: an **encoder** and a **Message Passing Neural Network** (MPNN). The encoder extracts features from nodes and edges, transforming the input data into a format suitable for relational reasoning. The MPNN propagates information across the graph, enabling the model to capture and leverage dependencies between nodes for better predictions.

To enhance the model's ability to focus on the target class during unlearning, a global feature vector, represented as the u vector, was introduced. This vector encodes high-level contextual information about the target class. By integrating this information, the model can effectively adjust its parameters to forget a specific class while maintaining its performance on the retained classes.

3.3 Integration of pre-trained graph with the Cora Dataset

To improve the model's understanding of graph structures, a pre-trained GMN was integrated with the Cora dataset. This approach leverages the pre-trained model's ability to reason about relationships and dependencies between nodes, providing a robust foundation for further learning and unlearning tasks.
As part of this integration, the u vector was one-hot encoded to represent the first class in the dataset (e.g., Neural Networks):

```
# Initialize u as a one-hot vector to indicate the "forget" class
g_cnn.u = torch.zeros(1, dataset.num_classes)
forget_class_idx = 0  # We want to forget class 1 (indexed at 0)
g_cnn.u[0, forget_class_idx] = 1.0  # Set the forget class
```

The dataset was further divided into retain and forget subsets based on class labels. Nodes belonging to the forget class (class index 0) were included in the forget dataset, while all other nodes were included in the retain dataset. The u vector served as a global indicator during training, signaling the forget class for the model. This division enabled a structured approach to unlearning, aligning the dataset's structure with the model's objectives.

3.4 Selective unlearning

The unlearning process in this project targeted the first class in the Cora dataset. This was achieved by adjusting the loss function with influence gradients, which identify the model parameters most affected by the target class. By penalizing the reliance on these parameters, the loss function guided the model to effectively ”forget” the target class while preserving its performance on the retained classes. This adjustment allowed the unlearning process to focus on the specific influence of the forgotten class, ensuring a more targeted and efficient optimization.

Balancing effective unlearning with overall model robustness emerged as a key challenge in this process. While the primary objective was to remove knowledge of the forgotten class, it was equally important to ensure that the performance on the retained classes remained intact. Influence-based adjustments played a pivotal role in addressing this challenge. By selectively penalizing parameters strongly influenced by the forgotten class, the unlearning mechanism minimized collateral damage to non-targeted classes and preserved the model's ability to generalize effectively across retained data.

Influence gradients were instrumental in guiding this process. They enabled **targeted forgetting**, where parameters strongly influenced by the forgotten class were penalized more heavily, ensuring efficient removal of the target class's knowledge. At the same time, this approach minimized collateral damage by preserving shared features between classes, preventing the unlearning process from inadvertently compromising performance on the retained data. Moreover, influence-guided adjustments enhanced optimization efficiency, reducing the number of steps required to achieve effective forgetting.

A comparison between **influence-based loss adjustment** and traditional **L2 regularization** employed in the model highlights their distinct yet complementary roles. Influence-based adjustment focuses specifically on parameters influenced by the data to be forgotten, tailoring the optimization process to undo their effects. In contrast, L2 regularization applies a global penalty to all parameters, serving as a general stabilization mechanism to prevent overfitting. While L2 regularization ensures overall model stability, influence-based adjustment ensures precise unlearning by targeting the relevant parameters.

Together, these methods provide a balanced approach to unlearning, combining task-specific optimization with general model robustness. The current strategy demonstrates the potential of leveraging influence-guided adjustments while laying the groundwork for exploring even more nuanced and effective unlearning methodologies in future work.

4 Evaluation of the unlearning algorithm and results

The effectiveness of the unlearning algorithm was evaluated through several metrics. **Accuracy** was measured on the retained and forgotten datasets to determine the model's performance in distinguishing between these two groups. Additionally, **Membership Inference Attack** (MIA) scores were used to assess the model's ability to "forget" specific samples. These scores provided insight into the extent to which the model had successfully removed its reliance on the forgotten class. Together, these evaluation methods offered a comprehensive understanding of the unlearning process and its impact on the model's overall performance.

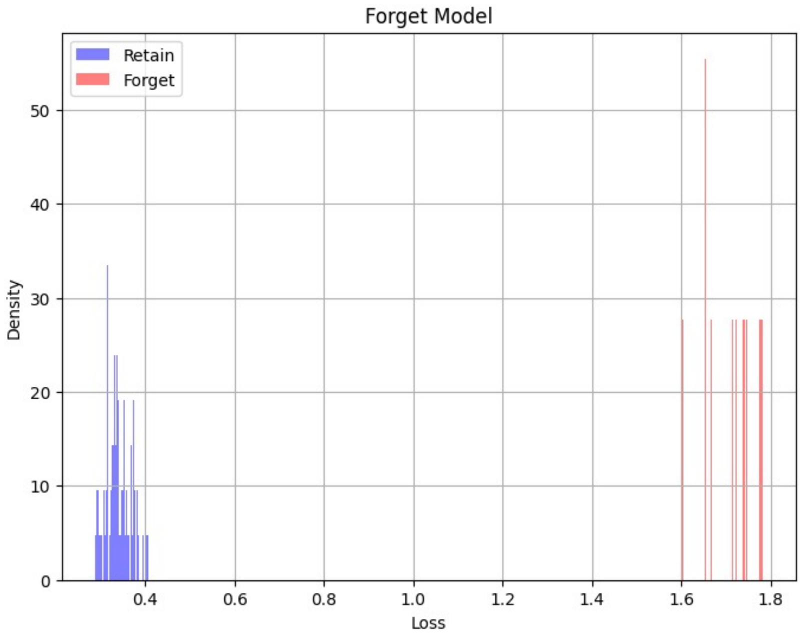

Figure 1: Forget Model

The output of the Forget Model evaluation is shown below:

```
Forget Model MIA Score: 1.0000
Retain Set Accuracy: 88.62%
Forget Set Accuracy: 37.18%
Test Set Accuracy: 77.86%
```

4.1 MIA score and forgetting

The Membership Inference Attack (MIA) score of 1.000 indicates that the unlearning algorithm was highly effective at distinguishing between retained and forgotten samples. This result demonstrates that the forgetting process worked as intended, with the model assigning markedly different behavior (measured through loss values) to the forgotten class compared to the retained ones. While this can be viewed as a success, such a perfect MIA score also suggests that the unlearning process may have been too aggressive. By excessively altering the model's parameters, the forgotten class becomes overly distinguishable, almost as if the model "overfits" to the task of forgetting, treating forgotten samples as clear outliers.

4.2 Accuracy metrics

The accuracy scores provide further insight into how the unlearning impacted the model's performance. The retain set accuracy of 88.62% demonstrates that the unlearning algorithm preserved the integrity of the retained classes, which aligns well with the primary goal of unlearning: removing specific knowledge without disrupting the rest of the model. This is a positive indication of the algorithm's ability to target only the intended class for forgetting.

However, the accuracy on the forgotten class dropped significantly to 37.18%, confirming that the model struggles to classify samples from this class post-unlearning. While this aligns with the goal of "forgetting" the class, it raises questions of whether a more refined approach could achieve subtler adjustments. The overall test accuracy of 77.86% shows that the model retains its generalization ability across the dataset, suggesting that the forgetting process did not catastrophically impair the model. Nevertheless, balancing forgetting and maintaining generalization remains a key challenge for further research.

4.3 Loss distribution analysis

The loss distribution plot clearly illustrates the separation between the retained and forgotten classes. Forgotten samples are assigned significantly higher loss values compared to retained samples, indicating that the model minimizes the importance of the forgotten class during decision-making. This separation is a strong indicator of the unlearning algorithm's effectiveness.

However, such a clear contrast may also reflect an over-aggressive forgetting process. By treating the forgotten class as outliers, the model might be sacrificing more subtle aspects of unlearning that could lead to smoother generalization. Ensuring a balance between effective forgetting and preserving overall model functionality remains an area for improvement.

4.4 Implications and future directions

The results highlight the potential of the unlearning mechanism but also reveal areas that require significant refinement. While the observed outcomes validate the feasibility of targeted unlearning, the unintended side effects, such as the perfect MIA score of 1.000, indicate that the process may have been overly aggressive. Future work could incorporate regularization techniques to prevent the model from over-penalizing forgotten samples, ensuring a more balanced adjustment. Additionally, expanding the evaluation metrics beyond MIA scores and accuracy could provide a more comprehensive understanding of the unlearning process and its broader implications.

A promising direction for future exploration involves forgetting random nodes from the network instead of targeting specific subsets such as classes. This approach could test the model's robustness to unstructured data removal and offer insights into how random node forgetting affects classification accuracy across study subjects. It would also simulate real-world scenarios where unpredictable data removal occurs, offering valuable lessons on model adaptability.

Further refinements to the approach could include enhancing edge attribute definitions to capture richer semantic relationships within the graph and experimenting with alternative loss adjustment strategies, such as class-weighted penalties. These improvements could lead to more nuanced and effective unlearning techniques, paving the way for robust, adaptable, and balanced solutions in graph-based learning.

References

[1] O. Hussein, "Graph neural networks series, part 4: The gnns message passing over-smoothing," Available at : https://medium.com/the-modern-scientist/graph-neural-networks-series-part-4-the-gnns-message-passing-over-smoothing-e77ffee523cc, 2023.

[2] K. Noda, "Ultimate guide to graph neural networks 1: Cora dataset," Available at : https://medium.com/@koki_noda/ultimate-guide-to-graph-neural-networks-1-cora-dataset-37338c04fe6f, 2022.