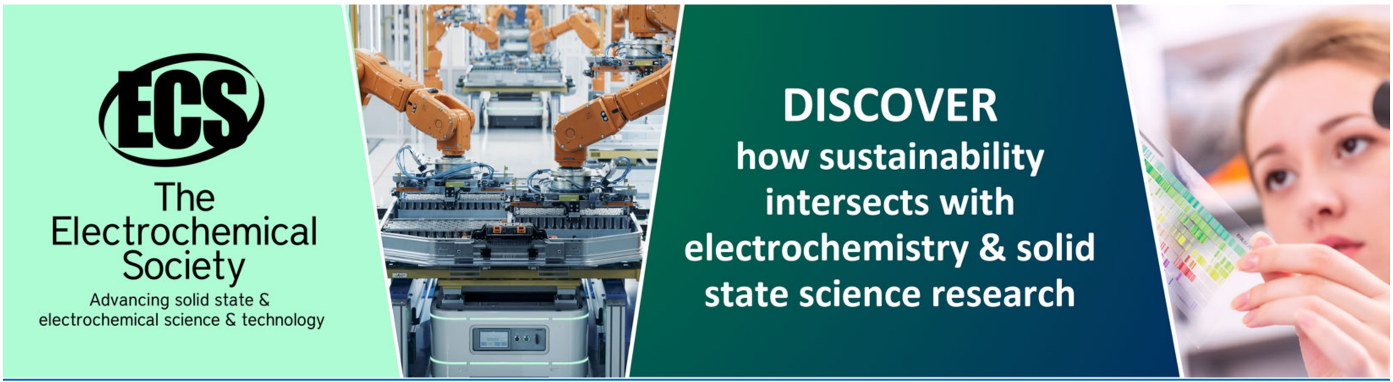
Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Unbalanced Class-incremental Learning for Text Classification Based on Experience Replay

To cite this article: Lifeng Chen *et al* 2023 *J. Phys.: Conf. Ser.* **2513** 012001

View the article online for updates and enhancements.


ECS
The Electrochemical Society
Advancing solid state & electrochemical science & technology
DISCOVER
how sustainability intersects with electrochemistry & solid state science research

Unbalanced Class-incremental Learning for Text Classification Based on Experience Replay

Lifeng Chen[1,a], Huaping Zhang[1,b], Silamu Wushour[2,c], Yugang Li[1,d]
[1] Beijing Institute of Technology, Beijing, China;
[2]Xinjiang University, Xinjiang, China;

[a]3120201008@bit.edu.cn; [b]kevinzhang@bit.edu.cn; [c]wushour@126.com; [d] lyg@bit.edu.cn

Abstract. While deep learning has achieved remarkable results for text classification, incremental learning for text classification is still a challenge. The main problem is that models suffer from catastrophic forgetting, which is they always forget knowledge learned before when labelled data comes sequentially and is trained in sequence. In this study, we propose methods of preventing catastrophic forgetting to handle unbalanced increased data. As an improvement over experience replay, our approaches improve the accuracy about 23.3% with 23% of all training data on Yahoo and 9.5% with 12% of all training data and on DBPedia.

1. Introduction

Incremental learning, also called continual learning, has been studied for decades[1]. According to [2], incremental learning can be described as three distinct scenarios. The first is the models of task-incremental learning (Task-IL), which always needs to be given the information that which task is to be performed. These models are trained with task-specific information because a task's identity is always provided. The second is domain-incremental learning (Domain-IL), where the identity of the task is not accessible at the time of testing. Compared to Task-IL, models only need to deal with the task at hand without requiring to infer which task it is. The third is class-incremental learning (Class-IL), which is similar to the commonly occurring problem in the real world of gradually learning new classes of data as they are added. For text classification, class-incremental models need to predict labels seen previously, and infer labels of the task at hand. In this paper, our method seeks to solve the third scenario.

Recently, it is seeing a surge of interest in this research community. To relieve catastrophic forgetting, several works have attempted to augment the loss function[3] that is being minimized during training to prevent model parameters, which are important to learning previous tasks, from significantly deviating their previous values, so that the models trained on new tasks perform well not only on new tasks, but also on previously learned tasks. Some other methods are based on an extra episodic memory, which is used to store data[3][4] or generate pseudo data of previous tasks[2]. Other methods combine regularization-based with replay-based methods[3].

All of the aforementioned methods use balanced training data, they do not consider the case of unbalanced data. In this study, we proposed our novel unbalanced class-incremental learning methods. To this end, considering the effectiveness of replay-based methods in most situations of incremental learning, we extended the methods of experience replay[4][5] to unbalanced class-incremental learning by (1) selectively storing previous data according to loss values, which is to select the most representative as possible, and reduce episodic memory usage, and (2) sampling and replaying examples from previous tasks with our sampling strategy when more and more new classes are deriving. Our contributions are as follows:

- We propose a new problem in which unbalanced classes are processed sequentially, which is more realistic for our application scenarios.
- Two strategies are used to extend experience replay to learn new tasks, which results in a method that outperforms traditional experience replay on the DBPedia and Yahoo datasets.
- Compared to mutil-task learning, our method uses less memory for storage and less training time, in contrast to experience replay, which uses more memory, especially for unbalanced data.

2. Related Work

Existing incremental learning research can be summarized into three main directions. The first is regularization-based methods, which make constraints on parameters of networks [3] to prevent them, especially important parameters, from changing too much while attempting to retain knowledge learned from previous tasks. Some are based on gradient projection [6][7]. To ensure the old tasks will not be affected, these methods keep the gradient updates from affecting the old tasks as much as possible.

The second is replaying data from an episodic memory module which stores previous data[3] [4] or generates pseudo data [2]. Experience replay (ER) is even effective in most of situations, even though a lot of methods have been proposed. Like it is mentioned above, replay-based methods always work well in most situations of incremental learning, so our approach is developed on ER.

Additional neural resource allocation is the third category, which prevents catastrophic forgetting by allocating different neurons or model parameters for different tasks [8][9]. It creates new neural resources as new tasks arrive[20], or simply creates a large network initially[10].

All those methods above have shown their effectiveness, but replay-based methods in most situations of incremental learning always work well. In this paper, we extend replay-based methods to solve the problem of the class-incremental learning for the unbalanced data.

In addition, recently continuous learning of some other NLP tasks has also tackled the problem of catastrophic forgetting. For example, continual learning models for sentiment analysis [6][11] have been proposed, dialogue slot filling[12], machine translation[13] and relation extraction[14].

3. Methodology

3.1. Problem Formulation

In this paper, we focus on class-incremental learning for a sequence of data (D_0, D_1, ..., D_{n-1}), where examples D_i(i = 1, ..., n)consist of classes that the model has never seen. For round i(R_i), examples include like Eq 1:

$$D_i = data\ of\ label_{ij}, (j = 1, \dots, m) \tag{1}$$

where j is the new label, m is the number of labels in D_i of R_i. And D_i is the unbalanced data.

3.1.1. Training and Testing data. In this class-incremental learning, we define the sequence of data as D = {D_0, D_1, ..., D_k}, D_i are the new classes comes in round i(R_i). Specially, for R_i, we train D_i and sample a small amount of data from the episodic memory. After D_i is trained, we test models on the same test data which includes all classes have seen or not or to see in the next rounds.

3.2. Our Methods

Our method is designed for real-world applications where input data is available sequentially, not simultaneously. It is naturally intrinsic, while humans constantly acquire and remember knowledge throughout our lives, but most computational models often suffer from catastrophic forgetting[15], which is the act of dramatically and rapidly forgetting knowledge learned from previous tasks while learning a new task. In situations like these, we have to retrain our input data to get an effective classifier. Since the computational cost of relearning previous knowledge is too high, a continual classifier is an absolute necessity for improving existing models. Our algorithm is shown in Algorithm 1. And for every training batch, we called it one step.

Algorithm 1 Strategies to Class-IL

```
Input: Training sets {$D_0$, …, $D_n$}, Memory M, Basic replay frequency $\alpha$, Thresh $\beta$
Output: Optimal model B, Updated memory M
   M = []
   Initialize B using pre-trained BERT
   for i=0; i<n; i++ do
      Buffer=[]
      if i!=0 then
         load B;
         load M;
         for batch in $D_i$ do
            train B;
            if step mod $\alpha$== 0 then
               for j=0; j<i; j++ do
                  sample every class averagely from the part of $D_j$ which is stored in M as a batch to replay;
               end for
            end if
            if loss of the batch > $\beta$ then
               store the batch into buffer temporarily;
            end if
         end for
      else
         for batch in $D_i$ do
           train B;
         end for
      end if
      store data in buffer into M;
      clear Buffer;
   end for
   return Model B, Updated M
```

3.2.1. Basic Model. In this study, our basic model is composed of a text encoder, a full-connected layer, a softmax layer and an episodic memory like [4]. Since Bert [16] is the state-of-the-art text encoder, which is based on the Transformer architecture[17].

3.2.2. Selective storing. Traditionally, we train all data available to get a classifier with high accuracy or train a great model for every task data. However, it costs too many resources like time and large memory to store the old data and models to do that. Here we propose a new method to select part of old data [18] instead of saving all data[4] into the episodic memory with tolerable accuracy lost. Moreover, data in the memory is supposed to include every class and as representative and diverse as possible. During training D_i, the model starts with the final parameter values from the former data learned, and learns new classes in the direction of gradient descent that make loss value as small as possible. As a result, we set a loss thresh manually to determine which data will store into the episodic memory. The loss is calculated as Eq 2:

$$L(\omega) = -\sum_{i=0}^{n} log\, p\,(y_i|x_i; W) \quad (2)$$

3.2.3. Average sampling. Assuming that the distribution of previous data (D_0, ..., D_{t-1}) is provided, for each experience replay, we replay t−1 batches for every previous task, so the replay frequency of every task is dynamic, which is flexible for tasks at hand according to the number of the tasks the model has learned. And there is no need to replay too frequently for the former tasks and too sparsely for the latter tasks. Importantly, we sample from the memory for every task and every class. With this module added, our methods work well even better on unbalanced data.

4. Experiments

4.1. Setup

4.1.1. Datasets. We evaluate baselines and our methods on two datasets. The results are shown in Table 1:

Table 1. Statistical information of datasets

Dataset	Type	Class
Yahoo	Question	10
DBPedia	Wikipedia	14

1. Split Yahoo: Yahoo is split into five subsets of two consecutive labels as {{0,1},...,{8,9}}. Also, due to the limited resources, we reduced the dataset shown in Table 2. For instance, in round zero(R_0), we add 2000 texts of class 0, 6000 texts of class 1; in round one(R_1), we add 2000 texts of class 2 and 6000 texts of class 3, etc. And we test models on the whole test data of Yahoo without reducing.

Table 2. Split Yahoo: for each round, two classes are added, they are 1000 examples and 3000 examples.

Round	Class added	Amount	Class added	Amount
0	0	2000	1	6000
1	2	2000	3	6000
2	4	2000	5	6000
3	6	2000	7	6000
4	8	2000	9	6000

2. Split DBPedia: DBPedia is split into seven subsets of two consecutive labels as {{0,1},..,{12,13}}. The detailed information is presented in Table 3.

4.1.2. Metrics. We train the newly added unbalanced data in combination with the episodic memory module which is used to store the selected previous data each round and test the trained model on the total test data. The total test data includes all classes, to look out how much knowledge models remember when classes increasing sequentially. Accuracy is used to evaluate them.

4.1.3. Implementation Details. We set the basic relay frequency α is 31, which means models replay previous data for every thirty steps in the experiments. And the value of the loss thresh is various for different datasets. In this study, we set 0.3 for split DBPedia and 0.5 for split Yahoo according to several tests.

Table 3. Split DBPedia: for each round, two classes are added, they are 1000 examples and 5000 examples.

Round	Class added	Amount	Class added	Amount
0	0	1000	1	5000
1	2	1000	3	5000
2	4	1000	5	5000
3	6	1000	7	5000
4	8	1000	9	5000
5	10	1000	11	5000
6	12	1000	13	5000

4.2. Baselines

In our experiments, the methods proposed in this paper are compared with the following baselines:

- Finetune[19]: the basic model is trained sequentially without any regularization on the loss function and the episodic memory.
- Muti-task Learning(MTL): the basic model is trained on all data available jointly. Considering it accesses to data from all tasks simultaneously, we use it as an upper-bound.
- Experience replay(ER)[4]: For this baseline, we train the basic model with the experience replay.

4.3. Results

We report the results of the final round in Table 4 on Yahoo dataset and DBPedia. From the results, our methods obviously outperform Finetune and ER. Especially, on DBPedia, the results on our methods are very close to the upper-bound(MTL) with saving part of data. For more details for the process of class-incremental learning, we test models after every round. The results are shows in the Table 5 and Table 6. Overall, whether Yahoo or DBPdia, our optimizations are effective.

Table 4. Results after all rounds on split Yahoo and split DBPedia. All results are average over 3 runs.

Method	Yahoo	DBPedia
Finetune	20.71	15.61
ER	16.62	88.74
Ours	**39.92**	**98.22**
MTL	64.60	98.66

Table 5. Results on split Yahoo. All results are average over 3 runs.

Method	R_0	R_1	R_2	R_3	R_4
Finetune	17.65	17.87	17.07	14.77	20.71
ER	17.65	18.03	31.35	24.30	16.62
Ours	17.65	**18.05**	**32.71**	**25.76**	**39.92**
MTL	17.65	28.76	42.18	51.38	64.60

Table 6. Results on split DBPedia. All results are average over 3 runs.

Method	R_0	R_1	R_2	R_3	R_4	R_5	R_6
Finetune	14.13	22.92	15.50	18.30	17.37	40.27	15.61
ER	14.13	27.88	35.37	50.43	67.02	77.45	88.74
Ours	14.13	**28.29**	**41.69**	**54.61**	**70.18**	**83.87**	**98.22**
MTL	14.13	28.09	42.17	56.02	70.41	84.62	98.66

To show the advantages of our methods compared to MTL, the records of time consuming are shown in Table 7. For R_i, the training time is about t for our methods, but it costs i*t for MTL. With selective storing, after all rounds, we just save the amount of 9,000 texts, and the total training texts is 40,000 for Yahoo. Besides, we finally save about 5,000 compared to 42,000 totally for DBpeia. That is, our methods only store about 23% of training examples on split Yahoo and 12% of training examples on split DBPedia, which means that our approaches significantly reduce the amount of data storage for continual learning.

Table 7. Time consuming on for each round on Yahoo and DBPedia. Time values fluctuate within 0.2 minutes above and below. (min)

Dataset	Method	R_0	R_1	R_2	R_3	R_4	R_5	R_6
Yahoo	Ours	1.89	1.98	2.16	2.22	2.28	-	-
	MTL	1.89	3.90	5.88	7.74	9.96	-	-
DBPedia	Ours	1.44	1.50	1.56	1.62	1.74	1.86	1.92
	MTL	1.44	2.94	4.38	5.88	7.32	8.82	10.32

The experiments above add the same proportions of class every round. To prove our methods can also be applied to the diverse proportions for different rounds, we add a small experiment here. The classes and amounts we added are shown in Table 8. And the results after last round are shown in the Table 9. From the results, our results are very close to the upper-bound MTL. Besides, though the performance of ER is also excellent, it stores all previous data to replay.

Table 8. Split DBPedia: for each round, two classes are increased.

Round	Class added	Amount	Class added	Amount
0	0	1000	1	4000
1	2	1000	3	6000
2	4	1000	5	8000
3	6	1000	7	10000
4	8	1000	9	12000
5	10	1000	11	14000
6	12	1000	13	16000

4.4. Ablation Study

In this section, we try to prove the effect of selective storage and balanced sampling of our methods. We only use each of them respectively each time and report their performance in Table 10, and the proportions of classes added for every round are the same as Table 3. From the results, we can see that removing average sampling leads to obvious performance degradation. However, without selective storing, for R_1, R_3, and R_4, it performs better. For R_4 and R_5, it gets worse results without selective storing than our methods. The accuracy decreases from 98.22 to 96.16. The reason is that when data is small, selective storing perhaps leads to the risk of overfitting the datasets. When data becomes large enough, it will have a better performance. Therefore, how to optimize the selective storing is a problem for us to study further in the future.

Table 9. Results with various ratios on split DBPedia. All results are average over 3 runs.

Method	R_0	R_1	R_2	R_3	R_4	R_5	R_6
Finetune	14.14	14.23	14.29	16.66	17.41	30.74	17.06
ER	14.14	28.11	41.42	55.76	69.58	83.92	97.77
Ours	14.14	**28.25**	**42.19**	**56.12**	**69.69**	**84.16**	**98.14**
MTL	14.14	28.33	42.34	56.23	70.48	84.42	98.28

Table 10. Ablation study results on split DBPedia. All results are average over 3 runs.

Method	R_0	R_1	R_2	R_3	R_4	R_5	R_6
Ours	14.13	28.29	**41.69**	54.61	70.18	**83.87**	**98.22**
-average sampling	14.13	27.54	37.43	49.94	68.21	79.77	90.46
-selective storing	14.13	**28.30**	41.25	**55.40**	**70.32**	83.52	96.16

5. Conclusion

In this study, we introduce a new approach to incremental learning of unbalanced classes with a small episodic memories and high accuracy. In addition, we do experiment using unbalanced data so that it is similar to real-world. We believe the proposed methods can be applied to all the experience replay scenarios. As for the selective storing, we have to study further for the optimal storing and avoiding the risk of overfitting when replaying the old data.

Acknowledgements

This research was funded by the Fundamental Enhancement Program Technology Area Fund (2021-JCJQ-JJ0059) and the Beijing Natural Science Foundation (4212026).

References

[1]. Barbara L McCombs. Motivation and lifelong learning. Educational psychologist, 26(2):117–127, 1991.
[2]. Generative replay with feedback connections as a general strategy for continual learning[J]. 2018.
[3]. Xu Han, Yi Dai, Tianyu Gao, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Continual relation learning via episodic memory activation and reconsolidation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6429–6440, 2020.
[4]. D'Autume C , Ruder S , Kong L , et al. Episodic Memory in Lifelong Language Learning[J]. 2019.
[5]. Schaul T , Quan J , Antonoglou I , et al. Prioritized Experience Replay[J]. Computer Science, 2015.
[6]. Ke Z , Xu H , Liu B . Adapting BERT for Continual Learning of a Sequence of Aspect Sentiment Classification Tasks[J]. 2021.

[7]. Wenpeng Hu, Qi Qin, Mengyu Wang, Jinwen Ma, and Bing Liu. Continual learning by using information of each class holistically. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 7797–7805, 2021.

[8]. Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pages 7765–7773, 2018.

[9]. Zixuan Ke, Bing Liu, and Xingchang Huang. Continual learning of a mixed sequence of similar and dissimilar tasks. Advances in Neural Information Processing Systems, 33:18493–18504, 2020.

[10]. Rusu A A , Rabinowitz N C , Desjardins G , et al. Progressive Neural Networks[J]. 2016.

[11]. Qi Qin, Wenpeng Hu, and Bing Liu. Using the past knowledge to improve sentiment classification. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1124–1133, 2020.

[12]. Yilin Shen, Xiangyu Zeng, and Hongxia Jin. A progressive model to enable continual learning for semantic slot filling. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1279–1284, 2019.

[13]. Shao C , Feng Y . Overcoming Catastrophic Forgetting beyond Continual Learning: Balanced Training for Neural Machine Translation[J]. arXiv e-prints, 2022.

[14]. Wu T , Li X , Li Y F , et al. Curriculum-Meta Learning for Order-Robust Continual Relation Extraction[C]// 2021.

[15]. Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In Psychology of learning and motivation, volume 24, pages 109–165. Elsevier, 1989.

[16]. Devlin J , Chang M W , Lee K , et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.

[17]. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

[18]. Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and M Ranzato. Continual learning with tiny episodic memories. 2019.

[19]. Yogatama D , D'Autume C , Connor J , et al. Learning and Evaluating General Linguistic Intelligence[J]. 2019.