CSN-300: LAB-BASED PROJECT

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MID-TERM REPORT

Gradient Projection Memory for Continual Learning

Under supervision of:
Prof. Pravendra Singh

Group Members:
Pranaw Raj[†] (19114064)
Divyansh Agarwal[†] (19115055)
Md Junaid Mahmood[†] (19116040)

March 23, 2022

[†]All members contributed equally to the project. All members worked under the supervision of Professor Pravendra Singh at the Department of Computer Science and Engineering, IIT Roorkee.

1 Overview

For our Lab based project, we have used a research paper published at ICLR-2021, as reference. **Gradient Projection Memory for Continual Learning** published by *Gobinda Saha*, *Isha Garg* and *Kaushik Roy* proposes a novel approach for continual learning to mitigate catastrophic forgetting.

In the subsequent sections, we present Introduction of the approach, the Proposed Algorithm, Experimental setup and Results of the algorithm over various datasets, Conclusion and our Future plans for the project.

2 Introduction

Continual Learning is a concept to learn a model for a large number of tasks sequentially without forgetting the knowledge obtained from the preceding tasks. One of the popular research areas in Continual Learning is the solution to the problem of Catastrophic Forgetting. In Neural Networks, Catastrophic Forgetting refers to the phenomenon of forgetting the information learned in the past tasks upon learning new ones. Contemporary solutions related to the problem of Catastrophic Forgetting are:

- One solution is to increase the size of the neural network. Methods in this category overcome catastrophic forgetting by dedicating different subsets of network parameters to each task. However, network-growth is a computationally expensive option and is not scalable.
- Another solution is to store episodic memories of old data. The idea is that the performance of old tasks is retained by taking gradient steps in the average gradient direction obtained from the new data and memory samples. However, these methods either compromise data privacy by storing raw data or utilize resources poorly, which limits their scalability.
- Another solution is to put constraints on the gradient updates so that task specific knowledge can be preserved. Adding the penalty term acts as a structural regularizer and dictates the degree of stability-plasticity of individual weight. However, due to such restrictions, performance suffers while learning longer task sequences.

The paper proposes a novel approach to tackle the problem of Catastrophic Forgetting in Continual Learning. As per the algorithm proposed in the paper, after learning each task, the entire gradient space of the weights is partitioned into two orthogonal subspaces: **Core Gradient Space (CGS)** and **Residual Gradient Space (RGS)**. Essentially, activations or representations of each layer form the bases of gradient subspaces. Using Singular Value Decomposition (SVD) on these activations, the minimum set of bases of the CGS is obtained. These bases ensure preservation of past knowledge and learnability for the new tasks. After learning each task, the bases are stored in a memory defined as the **Gradient Projection Memory (GPM)**. Any subsequent new task is learnt by taking gradient steps in the orthogonal direction to the space (CGS) spanned by the GPM. The approach is, thus termed as the **"GPM approach"**.

3 Algorithm

As shown in figure 1, the algorithm behind continual learning using Gradient Projection Memory(GPM) takes input in the form of a loss function, training datasets for all the tasks from 1 to T, learning rate α and threshold hyperparameter ϵ_{th}. Since the network has not been trained upon any task initially, Core Gradient Space (CGS) for each layer is set to be an empty array. Set of Core Gradient Space for all the layers in the neural network forms the Gradient Projection Memory (GPM). The weight matrix of the network is initialized to be W_0.

After all initialisation, the network is trained for each task. During training for any arbitrary task *t*, first of all a mini-batch from task *t* is sampled. Then, gradient descent is calculated considering this batch of training data. However, gradient descent update is done only along those directions that are orthogonal to the GPM (*CGS for each layer*). This process repeats until convergence is obtained.

After model has been trained over the task *t*, GPM needs to be updated in order to ensure preservation of past knowledge and learnability for the new tasks. For this, a mini batch of training data is again sampled.

Algorithm 1 Algorithm for Continual Learning with GPM

function TRAIN ($f_{\boldsymbol{W}}, \mathcal{D}^{train}, \alpha, \epsilon_{th}$)
Initialize, $\boldsymbol{M}^l \leftarrow [\,]$, for all $l = 1, 2, ...L$ // till L-1 if multi-head setting
$\mathcal{M} \leftarrow \{(\boldsymbol{M}^l)_{l=1}^L\}$
$\boldsymbol{W} \leftarrow \boldsymbol{W}_0$
for $\tau \in 1, 2, ..., T$ **do**
  repeat
    $B_n \sim \mathcal{D}_\tau^{train}$ // sample a mini-batch of size n from task τ
    gradient, $\nabla_{\boldsymbol{W}} L_\tau \leftarrow \text{SGD}(B_n, f_{\boldsymbol{W}})$
    $\nabla_{\boldsymbol{W}} L_\tau \leftarrow \text{PROJECT}(\nabla_{\boldsymbol{W}} L_\tau, \mathcal{M})$ // see equation (6, 7)
    $\boldsymbol{W} \leftarrow \boldsymbol{W} - \alpha \nabla_{\boldsymbol{W}} L_\tau$
  until convergence

  // Update Memory (GPM)
  $B_{n_s} \sim \mathcal{D}_\tau^{train}$ // sample a mini-batch of size n_s from task τ
  // construct representation matrices for each layer by forward pass (section 5)
  $\mathcal{R}_\tau \leftarrow \text{forward}(B_{n_s}, f_{\boldsymbol{W}})$, where $\mathcal{R}_\tau = \{(\boldsymbol{R}_\tau^l)_{l=1}^L\}$
  for layer, $l = 1, 2, ...L$ **do**
    $\hat{\boldsymbol{R}}_\tau^l \leftarrow \text{PROJECT}(\boldsymbol{R}_\tau^l, \boldsymbol{M}^l)$ // see equation (8)
    $\hat{\boldsymbol{U}}_\tau^l \leftarrow \text{SVD}(\hat{\boldsymbol{R}}_\tau^l)$
    $k \leftarrow \text{criteria}(\hat{\boldsymbol{R}}_\tau^l, \boldsymbol{R}_\tau^l, \epsilon_{th}^l)$ // see equation (9)
    $\boldsymbol{M}^l \leftarrow [\boldsymbol{M}^l, \hat{\boldsymbol{U}}_\tau^l[0:k]]$
  end for
end for
return $f_{\boldsymbol{W}}, \mathcal{M}$
end function

Figure 1: The proposed algorithm for the Continual Learning using **Gradient Projection Memory**. Each task is learnt by taking gradient directions orthogonal to the Core Gradient Space (CGS) for each layer. After learning each task, GPM is also updated to preserve task specific knowledge and ensure learnability of new tasks

For this mini-batch, representation matrix or activation matrix is calculated at each layer and using Singular Value Decomposition (SVD) significant bases for the matrix corresponding to this task are identified. The obtained important bases are then added into the Core Gradient Space (CGS) of that layer. The process repeats for each and every layer in the neural network and in this manner GPM gets updated. Core Gradient Space for each layer after learning the final tasks gives the final Gradient Projection Memory (GPM) for the network.

4 Experimental Setup and Reproduced results

We used two datasets for training and evaluating our model, namely- PMNIST and 10-Split CIFAR-100 dataset.

PMNIST: It is a variant of MNIST dataset (which contains 70000 grayscale images of digits from 0 to 9). Only difference here is that 10 sequential tasks are created using different permutations where each task contains 10 classes. Here, each task is considered as a random permutations of the original MNIST pixels.

10-Split CIFAR-100: This dataset consists of 60000 32x32 colored images in 100 classes, with 600 images per class. These 100 classes of CIFAR-100 are split into 10 tasks, each with 10 classes.

Training Details: All models are trained with plain Stochastic Gradient Descent (SGD).

Performance Metrics: The model is evaluated using two methods: **ACC** and **BWT** (Backward Transfer). The *ACC* metric is used to evaluate the classification performance. It is the average test classification of all tasks. Whereas, *BWT* is used to measure forgetting i.e., it indicates the influence of newer learning on past knowledge. Negative *BWT* indicates catastrophic forgetting.

$$ACC = \frac{1}{T}\sum_{i=1}^{T} R_{T,i}, \quad BWT = \frac{1}{T-1}\sum_{i=1}^{T-1} R_{T,i} - R_{i,i} \tag{1}$$

where T is the total number of tasks learnt up-to current point of evaluation and $R_{T,i}$ is the accuracy of model on i_{th} task after learning T_{th} task sequentially.

	PMNIST	Split CIFAR-100
No. of Tasks	10	10
Input size	1 x 28 x 28	3 x 32 x 32
No. of classes/tasks	10	10
No. of Training samples/tasks	54,000	4,750
No. of Validation samples/tasks	6,000	250
No. of Test samples/tasks	10,000	1,000

Figure 2: Different statistics related to two datasets - PMNIST and 10-Split CIFAR-100, which was used during evaluation of the model on various contemporary methods for continual learning.

	PMNIST	Split CIFAR-100
No. of Epochs	5	200
Batch Size	10	64
Model	FC layer with 2 hidden layer of 100 units each	5 Layer AlexNet

Figure 3: Details of number of epochs, batch size and model of the two datasets - PMNIST and 10-Split CIFAR-100 respectively.

4.1 Results for PMNIST

- Value of ϵ_{th} used is 0.95 for first layer and 0.99 for other subsequent layer.
- Our accuracy = 93.07, Accuracy in Paper = 93.91
- Our BWT = -0.0388, BWT in Paper= -0.03

Methods	Accuracy (%)	Backward transfer (BWT)
OGD	82.56	-0.14
OWM	90.71	-0.01
GEM	83.38	-0.15
A-GEM	83.56	-0.14
ER_Res	87.24	-0.11
EWC	89.97	-0.04
GPM	**93.07**	**-0.0388**

Figure 4: Comparison of results of proposed GPM method with that of other contemprory methods while training for PMNIST.

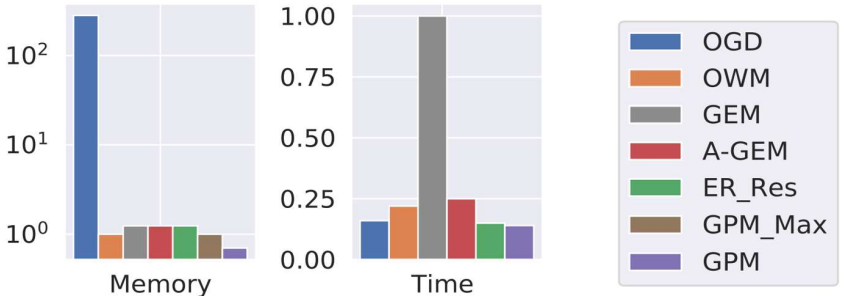

Figure 5: First graph indicates memory utilization by different methods while training for PMNIST. Second graph represents the per epoch training time for PMNIST tasks for different methods. .

4.2 Results for Split CIFAR-100

- Value of ϵ_{th} is 0.97 for all the layers and increasing it's value by 0.003 for each new tasks.

Methods	Accuracy (%)	Backward transfer (BWT)
OWM	50.94	-0.30
EWC	68.80	-0.02
HAT	72.06	-0.00
A-GEM	63.98	-0.15
ER_Res	71.73	-0.06
GPM	**71.77**	**-0.00**

Figure 6: Comparison of results of proposed GPM method with that of other contemprory methods while training for Split CIFAR-100.

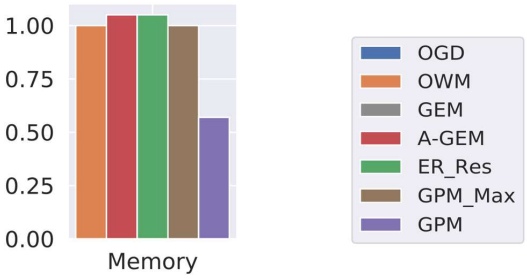

Figure 7: The graph indicates memory utilization by different contemporary methods for continual learning during training on Split CIFAR-100.

4.3 Effect of thresold hyperparameter (ϵ_{th})

In the proposed GPM continual learning approach, the term ϵ_{th} controls it's stability-plasticity dilemma. In case when ϵ_{th} is near to 0, optimizer will be allowed to change the weight along the directions of significance of past data – which may lead to interference (catastrophic forgetting). Whereas, in case when ϵ_{th} is close to 1, it will lead to low interference but it may affect the learnability of the new task.

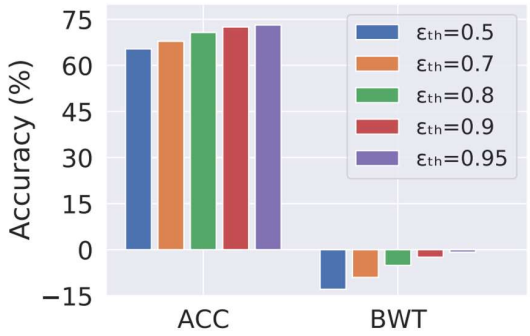

Figure 8: Effect of varying ϵ_{th} on the *ACC* and *BWT* of the model.

5 Conclusion

Till mid-term, we worked on understanding the research paper thoroughly, and re-produced the results using code on two datasets - PMNIST and Split CIFAR-100. We found that our results are very close to the results given in the paper.

We observe that Continual Learning using Gradient Projection Memory outperforms other contemporary solutions in terms of memory utilisation and accuracy over the learnt tasks. Thus, it effectively solves the problem of Catastrophic Forgetting.

However, for some datasets (e.g. *5-Datasets, as mentioned in the original paper*), Regularization-based methods achieve better performance as compared to GPM. Thus, using hybrid approaches like combining GPM with some data cleaning mechanism (*that extracts only relevant data necessary for learning a particular task so that only those bases contribute to the CGS that are important for that task*) opens scope for future exploration.

6 Future Work

We prepared a presentation, based on our understanding and presented it to our supervisor-Prof. Pravendra Singh.

Next, we plan to improve the method and get better results. For this, our plan is to create a hybrid model, such as using only processed data in training a model. We also plan to study and incorporate Model-Agnostic Meta-Learning for improving our existing model. We may also try some other approach or techniques depending upon its feasibility and time constraints. We plan to work on the above mentioned improvements for our final submission.

7 Acknowledgment

We thank Prof. Pravendra Singh for providing us with an opportunity to work on the project. References used for the work done till now:

- Saha, Gobinda Garg, Isha Roy, Kaushik. (2021). Gradient Projection Memory for Continual Learning.
- Reza Bagheri. (2020). Understanding Singular Value Decomposition and its Application in Data Science.
- Paul Hand. (2020). Continual Learning and Catastrophic Forgetting

Approved: _____

Professor Pravendra Singh
Department of Computer Science and Engineering, IIT Roorkee