Replicating Application of Elastic Weight Consolidation to Scrambled MNIST

Oliver Spohngellert

Abstract

Catastrophic forgetting refers to the tendency of Neural Networks to "forget" previously learned information when they are trained on new information. In "Overcoming Catastrophic Forgetting in Neural Networks", the authors propose Elastic Weight Consolidation as a method for mitigating catastrophic forgetting. In this work, we attempt to replicate the results of the MNIST experiments in the aforementioned paper, specifically plots 2 A-C. We succeed in replicating plots 2A and 2C, but are not able to completely replicate the results of plot 2B. Thus, we are able to show the efficacy of Elastic Weight Consolidation as a method for mitigating catastrophic forgetting.

1 Introduction

Catastrophic forgetting is a common problem for Deep Learning practitioners. Catastrophic forgetting occurs when a neural network is trained on multiple tasks sequentially, and the network "forgets" how to perform earlier tasks, resulting in poorer performance. This problem has been mentioned as early as 1989 in "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem" [1]. One proposed solution to this problem is the use of pretraining. In "Catastrophic Interference is Eliminated in Pretrained Networks", the authors demonstrate that using a pretrained network helps reduce error on sequential tasks [2]. More recently, it has been shown that the use of Generative Replay can alleviate the problem, but it has the drawback of linearly increasing training time with the number of former tasks [3].

In "Overcoming catastrophic forgetting in neural networks", the authors attempt to solve this problem by introducing Elastic Weight Consolidation (EWC) [4]. EWC works by adding a regularization term that penalizes the network for straying too far from important weights in the network. This is done as follows:

$$L(\theta) = L_B(\theta) + \frac{\lambda}{2}\sum_i F_i(\theta_i - \theta^*_{A,i})^2$$

Where L_B refers to the loss for task B (the current task), F_i refers to the diagonal of the fisher information matrix for weight i, and $\theta^*_{A,i}$ refers to the ith weight of the previous task A. This solution to catastrophic forgetting comes with the benefit of not requiring major changes to training procedure, robustness, and being fairly easy to implement.

In order to prove the efficacy of EWC, the authors conducted experiments when using EWC on the MNIST dataset. In these experiments, they trained neural networks on the MNIST dataset, where the pixels in the images have been scrambled, and they produced the following figure:

The main goal of this work was to reproduce the above figure.

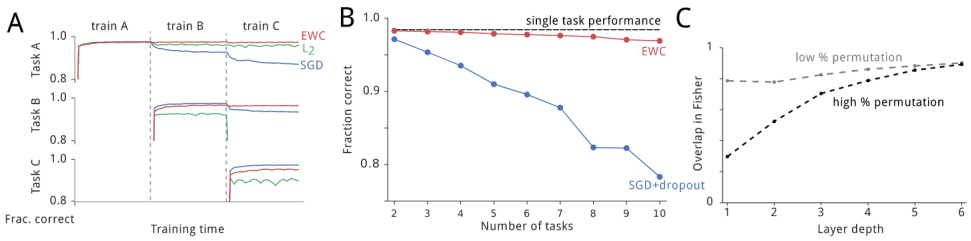


Figure 1: The figure produced in the original paper

2 Implementation

In order to replicate the figure in the original paper, EWC, L2 regularization by previous weights, random scrambling, calculation of the fisher diagonal, and training code had to be implemented. The full code can be found here. Below we will describe the general implementation approach.

2.1 EWC and L2

The fisher diagonal was calculated using the empirical fisher method, whose formula can be found in "Limitations of the Empirical Fisher Approximation for Natural Gradient Descent" [5]. Once the fisher diagonal was calculated on the training set, this was passed to a class called "EWC", along with a replica of the model after training on the task. These together were used to calculate the regularization term in EWC using PyTorch operations in order to allow for differentiation. L2 had a similar implementation but only used the previous nets and no fisher matrix. The only difference we had from the original paper was using the norm of the regularization sum (adding a square root term), as we found this allowed for a greater range of lambda values to be used:

$$L(\theta) = L_B(\theta) + \frac{\lambda}{2}\sqrt{\sum_i F_i(\theta_i - \theta^*_{A,i})^2}$$

The implementation of these classes, along with the scrambling code can be found here.

2.2 Training Code

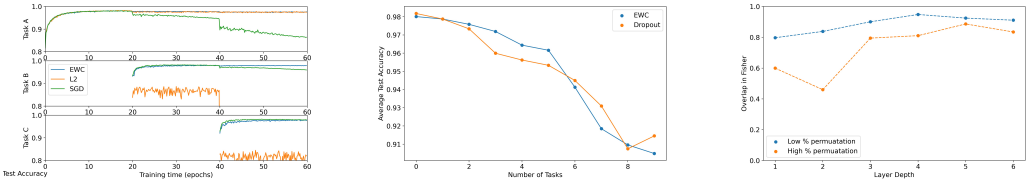
For each plot we implemented separate training loops, as different information needed to be saved for each, and they each used different network architectures. The majority of the hyperparameters used in the original paper were detailed, which allowed for a relatively painless implementation. However, the batch size and λ hyperparameters were not listed. For batch size, we assumed a batch size of 1, and used this when possible, but for plots B and C batch size was increased to save computation time. λ was set to 0.01 for plots A and C, and 0.01 or 0.005 for plot B. Further, the hyperparameters for plot B included a range of values, and the authors said they used a search for the optimal values. To save time, we used the middle value (1200) for hidden layer size, and the max value (10^{-3}) for learning rate. The implementations of training loops for each plot can be found at the following locations: Plot A, Plot B, Plot C.

2.3 Plotting

Once the data was saved, re-loaded and plotted. For plot C, the metric for overlap in fisher was "Fréchet Distance". The formula for this can be found in section 4.3 of the original paper, and this formula was used. The implementation of these plots can be found here.

3 Results

As can be seen in Figures 2a and 2c, both plots A and C could be replicated to a high level of accuracy. In Figure 2a, we can see that EWC outperforms SGD on taskA once training on task B begins training,



(a) (b) (c)

Figure 2: Replication of the 3 MNIST Plots

and the same could be said for tasks B and C. Further, L2 never achieves good performance on tasks B or C, the same as in the original paper. In addition, although the results in 2c are not exactly the same as the original paper, the high permutation task has lower overlap in earlier layers, then catches up in the later layers. The overlap is also generally higher in later layers than earlier layers.

However, as we can see in 2b, EWC seems to achieve siilar performance to Dropout after task 6. This is both because Dropout performs better than the results listed in the paper, and because EWC performs worse. It was hypothesized that the λ value picked for EWC was too high, forcing the model to overprioritize earlier tasks. To remedy this, the same experiment was tried with $\lambda = 0.005$.

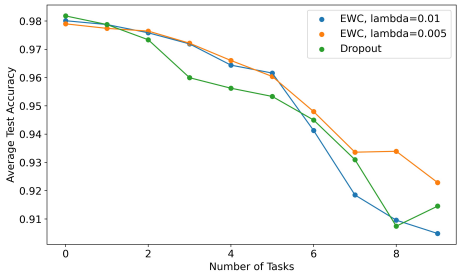


Figure 3: Plot B with both $\lambda = 0.01$ and $\lambda = 0.005$

As can be seen in Figure 3, using the lower value for λ did allow EWC to achieve better performance than Dropout, but only to a small degree. Though this does not fully replicate the plot referenced in the original text, it does display the importance of tuning λ. Here we can see that tuning this hyperparameter can greatley impact performance.

4 Discussion

Based on these results, it appears that EWC is a valid method for preventing catastrophic forgetting. More work would have to be done to determine how effective it is compared to other methods. Further, one important area of exploration would be speed comparisons. The code written for this project was not fully optimized for speed, and thus it is likely improvements could be made in this area.

References

[1] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989.

[2] K. McRae and A. Hetherington. Catastrophic interference is eliminated in pretrained networks. 1993.

[3] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Proceedings of the 31st International Conference on Neural Information*

Processing Systems, NIPS'17, page 2994–3003, Red Hook, NY, USA, 2017. Curran Associates Inc.

[4] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks, 2017.

[5] Frederik Kunstner, Lukas Balles, and Philipp Hennig. Limitations of the empirical fisher approximation for natural gradient descent, 2020.