Freezing Networks: Weight Preservation Procedure for Continual Learning

Davit Soselia, Levan Shugliashvili, Shota Amashukeli, Irakli Koberidze, Giorgi Chelidze
San Diego State University Georgia Artificial Intelligence and Machine Learning Club
Tbilisi, Georgia
`{dsoselia, lshugliashvili4659, samashukeli, ikoberidze,gchelidze}@sdsu.edu`

Abstract

We propose a new continual learning framework that uses a parallel neural network structure to emulate long-term and short-term memory by finding and conditionally freezing weights that have empirically proved most important to the network's performance on previously encountered tasks. Tests on MNIST and modified Character Font Images Data Set show that our method successfully performs above the baseline and avoid the problem of catastrophic forgetting.

1 Introduction

In recent years, neural networks have successfully been employed in a variety of tasks, including computer vision, data classification, and data generation. However, a major limitation of the traditionally employed methods has been the need for the network to have access to data representative of the whole distribution of the input feature space right from the start of the training, since sequentially training on isolated parts of the dataset leads to catastrophic forgetting - readjustment of weights in such way that the performance on previously encountered segments of the dataset drops drastically. This property of artificial neural networks - the ability to quickly "forget" previously learned classes in favor of newly learned classes - is employed in transfer learning, where training is done on a problem for which data is abundant and the network is later readjusted for the task at hand. However, as a consequence of this property, current approaches are unable to successfully solve problems that require classes to be added over time. This problem has been dubbed continual learning or lifelong learning [3][13] and presented as a major challenge in recent publications [10]. Development of a method or an architecture capable of such mode of learning is essential to the creation of decision making agents in general environments.

Several methods have been recently proposed, ranging from current-task oriented to old knowledge retention-oriented. We propose a parallel network structure in which one unit is responsible for memory consolidation and the other is responsible for adjusting to new tasks. This approach synthesizes several previous approaches in order to achieve compromise between the two extreme ends of the spectrum, and is inspired by observing human brain, where long and short term memory are used for continual learning without dependence on an increasing number of neurons [8] [4].

2 Related Work

Several solutions have been proposed to alleviate the problem of catastrophic forgetting. They tend to rely on a compromise between computational efficiency, memory complexity, and performance.

One recently introduced solution is the Progressive Networks - a model that offers high resistance to catastrophic forgetting through the creation of a parallel network for each new task and the complete freezing of the column networks trained on the past tasks [10]. This approach offers high overall

Preprint. Work in progress.

accuracy on all tasks, which comes at the expense of complexity and scalability. Another proposed approach that prioritizes memory retention over scalability introduces task-specific parameters [11]. This approach minimizes loss of information, but introduces a problem of choosing the correct weights. Some approaches, such as Learning Without Forgetting [6], are limited in their applicability to certain tasks, particularly to Reinforcement Learning.

Some approaches attempt rehearsals of the old tasks either by explicitly storing representative samples, or, in relatively recent models, by employing generative models for pseudo-rehearsal [9] [7].

A method that utilizes one-shot learning-like strategy for general sequential learning of large numbers of tasks has been attempted in One-Shot Imitation Learning [2].

An approach based on regularizing learning using elastic weight consolidation has been proposed [3]. This approach has been expanded upon in Progress & Compress [11]. This framework offers significant applicability to lifelong learning.

3 Freezing Procedure

The Freezing Network architecture uses two parallel networks. One is used for short term and the other for long term memory. Long term performance is achieved using selective weight freezing in one of the networks. Short term memory is implemented through a supplemental parallel network that is trained on each task with traditional methods without re-initialization.

3.1 Important weight identification

Our method attempts to empirically find the most important weights in the network through a procedure inspired by the Average Perturbation Sensitivity approach discussed in the Progressive Neural Networks paper [10]. We've also attempted a weight importance calculation procedure based on gradients, but the empirical approach outperformed it. The empirical procedure involves, upon the completion of the training on a given task, repeatedly setting a randomly picked selection (of size determined by a hyperparameter) of the long-term network's weights to zero and evaluating and assigning to each weight the absolute difference between the loss obtained in such way and the loss of the long-term network at the end of the training. The score for each weight is the average loss difference of each procedure repetition in which that given weight was selected to be set to zero. We opted for 200 repetitions of the procedure at the end of each training stage. Over multiple tasks, the weights' scores were set to be the averages of the averaged scores obtained this way.

3.2 Weight freezing and modification

At the end of the training stage on each task we save the long-term network's weights. We then freeze the top-d% (optimal value for d was found to be around 70%) most important weights of the long-term network, by reloading the previously-identified important weights after the training on each subsequent task is finished.

In practice, we do the above operation for each network layer's weights using matrix operations:

$$\mathbf{W_{new}} = \mathbf{W_{adjusted}} \cdot \mathbf{S} + \mathbf{W_{saved}} \cdot (\mathbf{J_n} - \mathbf{S})$$

where $\cdot$ signifies element-wise multiplication; $W_{\mathbf{adjusted}}$ is the weights matrix as modified by training on the most recent task; $\mathbf{W}_{saved}$ is the matrix containing the values of weights saved before the training on the most recent task; $\mathbf{S}$ is a matrix of the same size as W with elements corresponding to positions of top-d% elements of scores matrix being 0 and all of the other elements being 1; and $\mathbf{J_n}$ is a weights-sized matrix of ones.

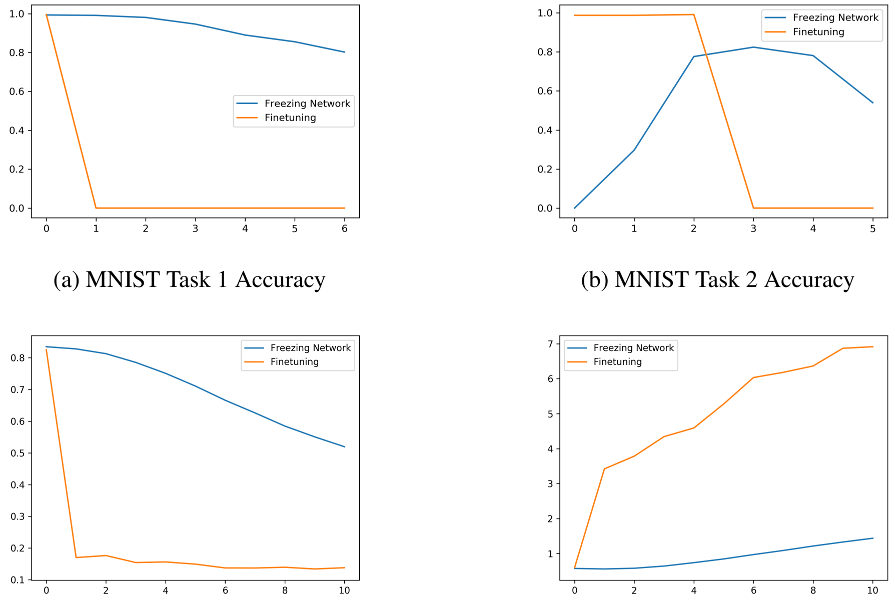
3.3 Proposed alternative: update slowing

An alternative approach to freezing the important weights would be to change the rate at which they re updated proportionally to their importance scores. For this, we propose the following update rule:

$$w_{new} = w_{saved} - (w_{saved} - w_{adjusted}) \cdot (1 - scaled\_importance)$$

Table 1: This table show the accuracy comparison between FN and Finetuning on MNIST after retraining on each new task.

Period	Freezing Network Accuracy %	Finetuning Accuracy %
End of Task 1	95.011	99.555
End of Task 2	57.714	0.01
End of Task 3	72.2600151171579	0



(a) MNIST Task 1 Accuracy

(b) MNIST Task 2 Accuracy

(c) Fonts Task 1 Accuracy

(d) Fonts Task 1 Loss

Figure 1: (a) and (b) show the network's prediction accuracy on the first MNIST task after a new task has been introduced, calculated after each epoch. (c) and (d) show the network's prediction accuracy and loss on the first task of the Character Font Image Data Set after the second subset has been introduced.

where w_{saved} is the value of the weight saved before the training on the most recent task, $w_adjusted$ is the value of the weight after an epoch of training on the most recent task, and $normalized\_importance$ is the the importance score of the weight (as calculated above) scaled from 0 to 1.

4 Preliminary Results

The training was done on several datasets, including MNIST[5], Character Font Images Data Set[1], and Georgian Hanwritten Characteres dataset [12]. Similarly to Progress & Compress, we evaluated the performance of our model in comparison to a network that has not implemented any protection against catastrophic forgetting (dubbed "finetuning") as a baseline[11].

We used MNIST divided into three parts. The first part contained digits 0 to 2, the second part contained digits 3-5, and the third part contained digits 6-9. The network was trained on each of the subclasses sequentially.

We used a similar approach with the fonts [1], but only divided these datasets into two parts. Classes that had less than 9000 samples were removed from the dataset, leaving 20 classes. We split the modified dataset into two parts, with 10 classes in each. Afterwards we repeated this with Georgian Characters dataset [12], and achieved similar results.

Preliminary results show that the model outperforms finetuning, and delivers performance comparable to Learning Without Forgetting, while requiring less computational power than Progressive Networks.

5 Summary and Future Work

This work introduced an approach to continual learning that uses a parallel network structure to implement long-term and short-term memory in order to avoid catastrophic forgetting; the long-term retention of performance on previous tasks in the long-term network happens through the identification and freezing of the weights that were most important to the previous tasks.

Modifications of the architecture and the procedures are planned for the immediate future, focusing on reconceptualizing important weight identification and implementing different procedures for utilizing the identified important weights in training. Extensive testing of the approach for Reinforcement Learning problems is planned. In addition, mechanisms for cross-talk between the long-term and the short-term parts of the network are being developed.

References

[1] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

[2] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1087–1098. Curran Associates, Inc., 2017.

[3] James Kirkpatrick, Razvan Pascanu, Neil C. 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. *CoRR*, abs/1612.00796, 2016.

[4] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *CoRR*, abs/1604.00289, 2016.

[5] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[6] Zhizhong Li and Derek Hoiem. Learning without forgetting. *CoRR*, abs/1606.09282, 2016.

[7] German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *CoRR*, abs/1802.07569, 2018.

[8] Marc Pickett, Rami Al-Rfou, Louis Shao, and Chris Tar. A growing long-term episodic & semantic memory. *CoRR*, abs/1610.06402, 2016.

[9] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning. *CoRR*, abs/1811.11682, 2018.

[10] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.

[11] Jonathan Schwarz, Jelena Luketina, Wojciech M. Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. *CoRR*, abs/1805.06370, 2018.

[12] Davit Soselia, Magda Tsintsadze, Levan Shugliashvili, Irakli Koberidze, Shota Amashukeli, and Sandro Jijavadze. On georgian handwritten character recognition. *IFAC-PapersOnLine*, 51(30):161–165, 2018.

[13] Ion Stoica, Dawn Song, Raluca Ada Popa, David A. Patterson, Michael W. Mahoney, Randy H. Katz, Anthony D. Joseph, Michael I. Jordan, Joseph M. Hellerstein, Joseph E. Gonzalez, Ken Goldberg, Ali Ghodsi, David Culler, and Pieter Abbeel. A berkeley view of systems challenges for AI. *CoRR*, abs/1712.05855, 2017.