# Elastic Weight Consolidation for Reduction of Catastrophic Forgetting in GPT-2

**Anonymous ACL submission**


## Abstract

Neural networks are naturally prone to the effects of catastrophic forgetting during fine-tuning. Despite the extensive adoption of transformers, little research has been done to investigate the effects of catastrophic forgetting on attention-based architectures. In this work, we used elastic weight consolidation (EWC) to mitigate catastrophic forgetting caused by fine-tuning in one of the foundation models, GPT-2. We show that by using EWC, we can significantly slow down the forgetting process without major penalty for the performance of the task model is fine-tuned for. We also determine that the majority of important weights is located in self-attention layers, and the parameters most sensitive to change are located in the normalization layers. Finally, we explore the instability of the EWC and potential performance issues.


## 1 Introduction

The neural network training process is usually split into two parts (Yosinski et al., 2014): pre-training on data representing some broad domain, and fine-tuning using a more specific data set. For NLP tasks, a model used for fine-tuning is usually a language model trained on some kind of a large data set. In this paper, we examine how such language model tends to forget prior broad knowledge when it is fine-tuned for a new, more specific task. The issue of catastrophic forgetting(McCloskey and Cohen, 1989). is caused by the changes of the pre-trained model's weights during a fine-tuning phase when the model is forced to learn a completely new set of data (Goodfellow et al., 2015). Surprisingly, little research was carried out regarding catastrophic forgetting effects on transformers' performance, especially regarding foundation models (Bommasani et al., 2021).

This work investigates the practical implementation and effects of Elastic Weight Consolidation (EWC) applied to a large-scale transformer GPT-2 that was pre-trained on large text corpora and fine-tuned using conversational data. We chose EWC as a method due to the fact how much interpretability and analysis options it provides.

## 2 Elastic weight consolidation for transformers

The change in model parameters caused by fine-tuning can be highly disruptive as neural networks' performance is quite sensitive to small perturbations in model parameters (Shu and Zhu, 2019).

It is important to note that for models with a smaller set of parameters, the problem of catastrophic forgetting can be attributed to the model's limited capacity. Bhattamishra (Bhattamishra et al., 2020) proposes that modern architectures such as transformers that contain millions or even billions of parameters will probably retain unused capacity.

If we think about catastrophic forgetting in terms of the model's weights deviation from its original values, we can use weight regularization to combat this issue. Regularization relies on keeping foremost weights as close as possible to original values while providing some range of motion for parameters that are not considered important for keeping prior knowledge intact (Hastie et al., 2009). The way importance is assigned to weights varies between methods. For example, in L2 regularization, all weights are equally important. EWC allows us to assign different importance values to model parameters based upon their contribution to prior task performance.

The original EWC method was proposed in the paper "Overcoming catastrophic forgetting in neural networks" (Kirkpatrick et al., 2017). As we plan to use EWC during fine-tuning, the training process shall be divided into two steps—a language modeling task and a conversational task. These tasks are semantically close to each other. Therefore during fine-tuning, the model can use relevant information from parameters important for the language modeling without significant alteration and perform most of the necessary parameter fitting on unnecessary weights. The constraint mechanism used to protect vital parameters for initial knowledge is implemented as a quadratic function using the Fisher information matrix (or FIM), hence the term elastic. During fine-tuning, we added an additional penalty to the loss function to enforce EWC.

## 3 Continual learning methods

There are several approaches to tackle the challenge of catastrophic forgetting, and they usually represent some form of parameter regularization (Parisi et al., 2019). We chose EWC because it allows us to store, use and analyze regularization data separately from the model. Moreover, the Fisher information matrix used in EWC offers a straightforward and meaningful way to analyze how memorized knowledge works and what relation it has to a type of parameters.

The Side-tuning method focuses on adding a side model to a pretrained base model to use present knowledge and added capacity for learning new skills. Usually, the same architecture or a lighter, distilled version is used as the side model (Zhang et al., 2020). Learning without forgetting (LWF) (Li and Hoiem, 2018) and Incremental moment matching (IMM) (Lee et al., 2017) are other efficient methods. For instance, the LWF method extends the base model by adding a small set of new parameters and a new output layer while the old output layer is preserved for regularization.

A more comprehensive review of strategies to combat catastrophic forgetting can be found in (Biesialska et al., 2020).

## 4 Datasets

We opted to test our approach on the GPT-2 transformer created by OpenAI (Radford et al., 2019). For the GPT-2 training, the original paper's authors used the WebText corpus comprised of 40GB of text collected from 8 million web pages. As the original WebText corpus has not been released yet and probably will not be released at all, community reconstruction called OpenWebText (Gokaslan and Cohen, 2019) (OWT) was used in our experiments. Different subsets of the OpenWebText were used during research: a general population with the size of 32GB, a sample of the general population for EWC calculation with the size of 1GB (randomly sampled), and a data set randomly sampled from the 1GB dataset with the size of 50MB that was used for perplexity calculation. Cascading subsets of OWT were chosen to keep computation within feasible limits.

To fine-tune the GPT-2 for the conversational task, we used the data set from Conversational Intelligence Challenge 2 (ConvAI2), the same one, Hugging Face team used for building the persona-oriented model (Persona-chat). The ConvAI2 PERSONA-CHAT data set (initially presented in (Zhang et al., 2018)) consists of around ten thousand dialogues crowdsourced using personality descriptions provided to participants as part of their character. The test sample covers around 6% of the PERSONA-CHAT data set.

## 5 Model Architecture and Training

The GPT-2 model was used as a baseline model to start. Fine-tuning was performed using pre-trained weights from the language modeling step, task A. For task B, the conversational fine-tuning task, we chose a persona-based conversational architecture by Hugging Face, identical to GPT-2 except for the next sentence prediction head. The sentence prediction head determines the correct sentence among distractors when the end-of-sequence token is passed using the cross-entropy loss function.

Perplexity on the OpenWebText test sample was chosen as the primary metric for catastrophic forgetting detection during and after GPT-2 fine-tuning. Accuracy and perplexity were also used to measure the quality of fine-tuning on the ConvAI2 test sample. To implement EWC during fine-tuning, we computed importance matrices on the 1GB of OWT data. The pre-trained model's parameters were used to measure the fine-tuned weights' deviation from original values.

Using calculated deviations and importance metrics, the EWC penalty can be added to the model's loss function with some coefficient. We used the coefficient of 1 as it showed a good balance between weights restraining and fine-tuning performance.

During the fine-tuning step, we used the AdamW optimizer with a Cosine Annealing Scheduler with a learning rate of 6.25e-6. We used NVIDIA DGX for fine-tuning and EWC calculations. We fine-tuned two models—with EWC and without—for 10 epochs.

| Epoch | Accuracy, PC test sample | | Perplexity, PC test sample | | Perplexity, OWT test sample | |
|---|---|---|---|---|---|---|
| | without EWC | EWC | without EWC | EWC | without EWC | EWC |
| 1 | 0.54 | 0.56 | 4.27 | 4.32 | 23.4 | 14.1 |
| 2 | 0.61 | 0.63 | 3.61 | 3.56 | 23.4 | 14.9 |
| 3 | 0.61 | 0.60 | 3.61 | 3.63 | 28.7 | 15.6 |
| 4 | 0.63 | 0.63 | 3.41 | 3.39 | 50.8 | 24.9 |
| 5 | 0.65 | 0.66 | 3.20 | 3.22 | 57.2 | 18.6 |
| 6 | 0.64 | 0.64 | 3.23 | 3.24 | 676.5 | 20.9 |
| 7 | 0.66 | 0.66 | 3.12 | 3.16 | 2203.8 | 21.1 |
| 8 | 0.66 | 0.67 | 3.07 | 3.09 | 4491.2 | 38.2 |
| 9 | 0.66 | 0.67 | 3.08 | 3.11 | 1737.3 | 32.1 |
| 10 | 0.67 | 0.67 | 3.03 | 3.06 | 14598.4 | 38.8 |

Table 1: Accuracy and perplexity dynamics during 20 epochs of training models with and without EWC

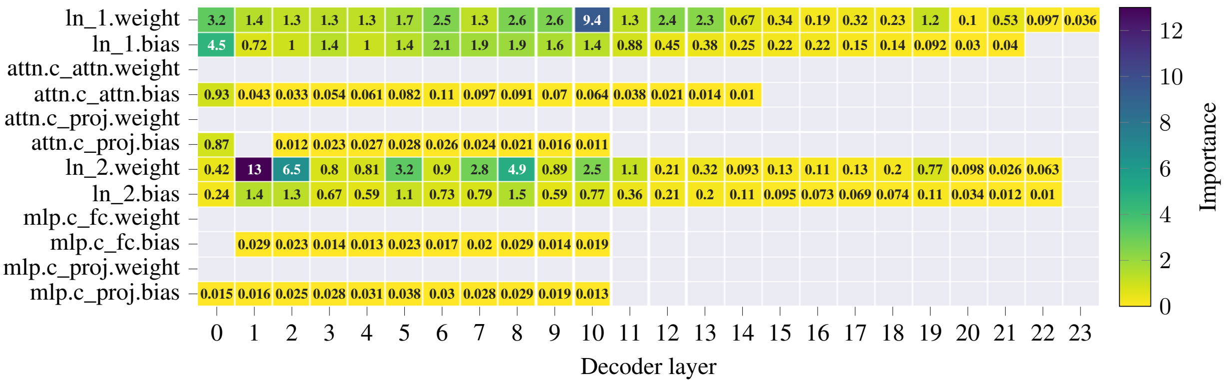


Figure 1: Mean metric of importance by layer and decoder level. Model outputs are the most sensitive to change in normalization layer parameters.

## 6 Results and Analysis

When looking into perplexity and accuracy metrics on Table 1 measured on the PERSONA-CHAT test sample, we can observe no big difference between the model with EWC and the model without EWC. The model with EWC shows a slightly bigger perplexity, which is expected, as a model capacity that was available for the model without EWC utilization is now restrained by an "elastic" penalty.

The model with EWC shows significantly lower perplexity on the OWT test sample (Table 1). Prior to the fifth epoch, both models show perplexity lower than 100. However, starting from the sixth epoch, magnitudes for models started to differ significantly: the model with EWC will never reach 500, while the model without EWC can achieve perplexity values up to 14598 (exact values can be found in Table 1). Though the absolute difference between perplexity values of the two models can look staggering, we have to account for the instability of the method and perplexity metric.

The perplexity metrics for models differ on the order of a few magnitudes—the model with the EWC shows significantly lower values for perplexity during all epochs. After the fifth epoch, perplexity for the model without the EWC penalty goes in the range of thousands. Penalized model perplexity values also grow, but this rate is moderate.

The model with the EWC penalty is on par with the plain model when considering metrics on the PERSONA-CHAT test set. However, this model is far better at remembering information from the OWT set.

### 6.1 Investigation of the matrix of importances

If we take a closer look at each decoder block's importance (values from FIM) for each weight matrix, we can see that most vital parameters are located on normalization layers of the decoder block and not on self-attention layers.

Normalization layers in figure 1 have massive gradient values because the slightest change in layer normalization will significantly change a model's output. Another reason for such a result is the difference in shapes—normalization layers have the smallest shape among other decoder layers. For example, `attn.c_attn.weight` has shape

| Threshold | 0.01 | 0.05 | 0.1 | 0.5 | 1.0 | 5.0 | 10 | 50 | 100 | 500 | 1000 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Number of parameters | 1764004 | 394843 | 225955 | 71615 | 37128 | 3070 | 1137 | 267 | 156 | 41 | 22 |

Table 2: Number of significant parameters by threshold. Number of important parameters falls as importance threshold rises.

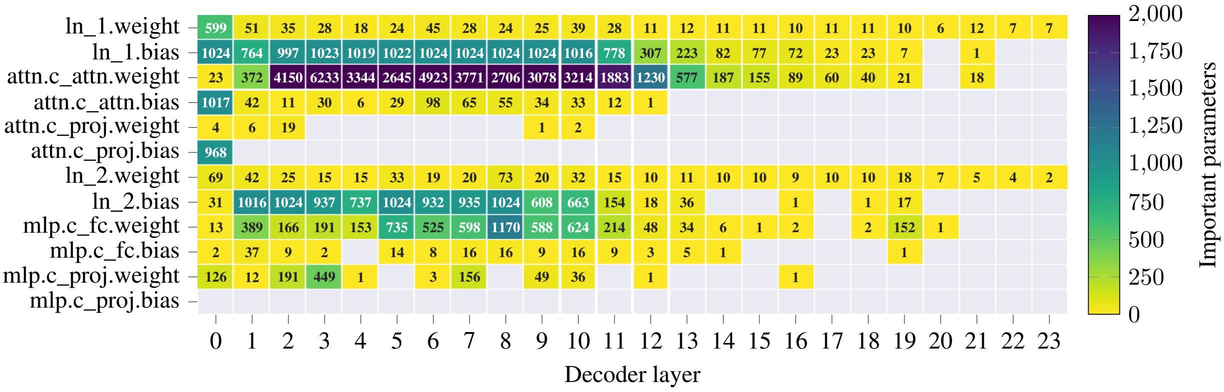


Figure 2: Number of parameters that has exceeded importance metric of 0.5 by layer and decoder level. Self-attention layers are also the most important in terms of number of parameters that surpass threshold 0.5.

of [1024, 3072] and `ln_1.weight` has a shape of [1024]. Suppose the normalization layer has 800 significant parameters and the self-attention layer has 80000. In that case, normalization layers will show a higher mean value than the self-attention layers. However, it is worth mentioning that self-attention layers have the largest number of important weights due to the sheer number of parameters in these layers.

Figure 2 shows how many parameters surpass the arbitrarily chosen importance threshold of 0.5. This image shows the most significant weights are primarily located in the main attention block `attn.c_attn.weight`. This value of 0.5 was chosen for representative purposes, and it does not affect the final result—self-attention layers always contain the most amount of important weights.

The number of important weights will decline if we increase the threshold. When the threshold reaches the values of hundreds, the number of important weights is almost non-existent if we keep in mind that the model contains several billions of parameters. Table 2 shows this dynamic using several thresholds.

## 7 Conclusion

During the analysis of EWC importance matrices, we found out that the most important weights are located in self-attention layers. We also determined that the most sensitive to change parameters are located in the normalization set of weights. Using EWC allows the GPT-2 to retain its knowledge acquired during pre-training and use it for continual learning. The nature of the EWC method enables in-depth analysis of important layers, sensitivity analysis. The fact that EWC matrices can be stored separately adds flexibility to the method.

### 7.1 EWC Limitations And Future Work

Despite all positives, EWC slows down training time and significantly increases memory consumption. The problem of extensive memory consumption can be critical regarding training large-scale transformers. Models that once fit on a single GPU will no longer do so when EWC is utilized.

Though we tried to produce comprehensive research, we can identify some areas for improvement. The first major improvement would be to increase the amount and quality of data used for fine-tuning and EWC calculation. Other transformer architectures in conjunction with EWC can be investigated, such as BERT, T5, or GPT-3.

## References


Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020. On the ability and limitations of transformers to recognize formal languages. *arXiv preprint*, arXiv:2009.11264.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In

*Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2015. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint*, arXiv:1312.6211.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc.

J. Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, J. Veness, G. Desjardins, Andrei A. Rusu, K. Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, C. Clopath, D. Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. volume 114, pages 3521–3526.

Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*, volume 30, pages 4652–4662. Curran Associates, Inc.

Zhizhong Li and Derek Hoiem. 2018. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, pages 109–165. Elsevier.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.

Persona-chat. Persona-chat dataset.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Hai Shu and Hongtu Zhu. 2019. Sensitivity analysis of deep neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4943–4950.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, volume 27, pages 3320–3328. Curran Associates, Inc.

Jeffrey O Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. 2020. Side-tuning: A baseline for network adaptation via additive side networks. *arXiv preprint*, arXiv:1912.13503.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.