Mitigating the Diminishing Effect of Elastic Weight Consolidation

Canasai Kruengkrai **Junichi Yamagishi**
National Institute of Informatics, Japan
{canasai,jyamagishi}@nii.ac.jp

Abstract

Elastic weight consolidation (EWC, Kirkpatrick et al. 2017) is a promising approach to addressing catastrophic forgetting in sequential training. We find that the effect of EWC can diminish when fine-tuning large-scale pre-trained language models on different datasets. We present two simple objective functions to mitigate this problem by rescaling the components of EWC. Experiments on natural language inference and fact-checking tasks indicate that our methods require much smaller values for the trade-off parameters to achieve results comparable to EWC.[1]

1 Introduction

New training data may arrive after we have spent considerable time training our model on the data at hand. A simple method for exploiting both new and old training data is to mix them and retrain the model from scratch. However, this mix-and-retrain method is neither always practical nor economical, especially in academic environments where computational resources are limited.

Sequential training is a potential alternative approach but faces a difficult challenge called *catastrophic forgetting* in which the performance on old data drastically drops when we train a model on new data. There exists a line of work that has addressed this challenge (Rusu et al., 2016; Li and Hoiem, 2018; Kirkpatrick et al., 2017; Mallya et al., 2018; He and Jaeger, 2018; Zhang et al., 2020). In this paper, we are particularly interested in elastic weight consolidation (EWC, Kirkpatrick et al. 2017), which has been shown to be helpful for domain adaptation (Saunders et al., 2019; Thompson et al., 2019).

EWC adds a regularization term to the objective function to ensure that the model works well on both new and old data. We empirically find that EWC requires unexpectedly large values for the trade-off parameter (λ) between the regularizer and the loss to be effective when applying to pre-trained language models. Figure 1 shows such a phenomenon in which EWC has no effect in preventing catastrophic forgetting of the prior dataset (MNLI) with λ in the range of $[10^0, 10^4]$. We have to scale λ up to $[10^5, 10^7]$, which is an unusual range of hyperparameters. To the best of our knowledge, this phenomenon has not been reported in the literature.

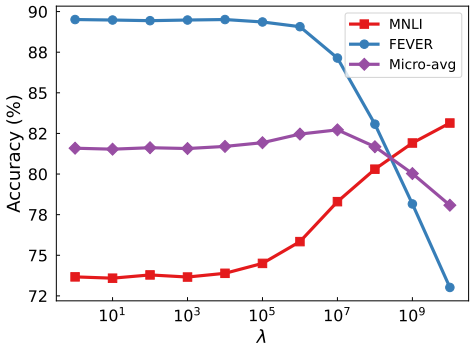


Figure 1: Accuracy vs. trade-off parameter λ. We sequentially fine-tune BERT (Bidirectional Encoder Representations from Transformers, Devlin et al. 2019) on MNLI (Williams et al., 2018) and FEVER (Thorne et al., 2018) and evaluate performance on the balanced dev sets. EWC starts to increase the accuracy of the prior dataset (MNLI) when increasing λ to 10^5 and yields the highest average accuracy at 10^7.

We propose two simple objective functions for mitigating the diminishing effect of EWC. Our objective functions rely on rescaling the components of EWC. Specifically, the first objective function involves taking the square root of the regularization term, while the second one involves using the absolute value of the gradient instead of the squared gradient. Both of our objective functions can reduce the values of the trade-off parameter λ by three to seven orders of magnitude while producing results similar to those of the original EWC.

[1]Our code is available at https://github.com/nii-yamagishilab/ewc.

Proceedings of the 29th International Conference on Computational Linguistics, pages 4568–4574
October 12–17, 2022.

2 Background

2.1 Problem formulation

We consider a supervised learning problem in which the task is to map an input $x \in \mathcal{X}$ to a label $y \in \mathcal{Y}$. We need to train a model $h_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^d$. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{M}$, we typically estimate $\boldsymbol{\theta}$ on the basis of empirical risk minimization (ERM, Vapnik 1992):

$$J_{\text{ERM}}(\boldsymbol{\theta}) = \frac{1}{M} \sum_{(x,y)\in\mathcal{D}} \mathcal{L}(h_{\boldsymbol{\theta}}(x), y), \quad (1)$$

where $\mathcal{L}$ is the negative log likelihood loss:

$$\mathcal{L}(h_{\boldsymbol{\theta}}(x), y) = -\sum_{y\in\mathcal{Y}} \mathbb{1}\{\hat{y} = y\} \log p_{\boldsymbol{\theta}}(\hat{y}|x).$$

Our base model $h_{\boldsymbol{\theta}}$ is a neural network containing a multilayer perceptron (MLP) on top of a pre-trained language model (e.g., BERT). Thus, we define $p_{\boldsymbol{\theta}}(\hat{y}|x) = \text{softmax}(h_{\boldsymbol{\theta}}(x))$, where $h_{\boldsymbol{\theta}} = \text{MLP}(\text{BERT}(x))$. The model parameters $\boldsymbol{\theta}$ include those in the MLP and BERT.

2.2 Elastic weight consolidation

Elastic weight consolidation (EWC, Kirkpatrick et al. 2017) is based on a Bayesian framework that seeks to approximate the posterior distribution of $\boldsymbol{\theta}$ conditional on two datasets. Let $\mathcal{D}$ and $\mathcal{D}^0$ denote the current and prior datasets, respectively. We express the posterior distribution as:

$$\begin{aligned} p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{D}^0) &= \frac{p(\boldsymbol{\theta}, \mathcal{D}, \mathcal{D}^0)}{p(\mathcal{D}, \mathcal{D}^0)}, \\ &= \frac{p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{D}^0)p(\boldsymbol{\theta}, \mathcal{D}^0)}{p(\mathcal{D}, \mathcal{D}^0)}, \\ &= \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D}^0)p(\mathcal{D}^0)}{p(\mathcal{D})p(\mathcal{D}^0)} \\ &\propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D}^0), \end{aligned} \quad (2)$$

where we assume that $\mathcal{D}$ and $\mathcal{D}^0$ are conditionally independent in the third line and ignore the constant in the last line. Taking the log on both sizes of Eq. (2), we have:

$$\log p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{D}^0) = \log p(\mathcal{D}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}|\mathcal{D}^0). \quad (3)$$

The first term on the right-hand side corresponds to the log likelihood of $\mathcal{D}$, which can be computed using Eq. (1). The second term is intractable but can be approximated using a second-order Taylor expansion of the KL-divergence around the parameters of the previously trained model, $\boldsymbol{\theta}^0$:

$$\log p(\boldsymbol{\theta}|\mathcal{D}^0) \approx \frac{1}{2}\Delta\boldsymbol{\theta}^\top \mathbf{H} \Delta\boldsymbol{\theta}, \quad (4)$$

where $\Delta\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\theta}^0$ and $\mathbf{H}$ is the expected negative Hessian of the posterior distribution (Pascanu and Bengio, 2014). Computing $\mathbf{H}$ is impractical. Kirkpatrick et al. (2017) proposed approximating $\mathbf{H}$ using the diagonal of the Fisher information matrix. Let $\text{diag}(\mathbf{f})$ be the diagonal matrix with diagonal $\mathbf{f}$. We estimate $\mathbf{f}$ with the average of the squared gradient across some N subsamples $\mathcal{S}^0$:

$$\mathbf{f} = \frac{1}{N} \sum_{(x,y)\in\mathcal{S}^0} \left(\nabla_{\boldsymbol{\theta}^0} \mathcal{L}(h_{\boldsymbol{\theta}}(x), y)\right)^2. \quad (5)$$

Replacing $\mathbf{H}$ with $\text{diag}(\mathbf{f})$, we can simplify Eq. (4) as:

$$\log p(\boldsymbol{\theta}|\mathcal{D}^0) \approx \frac{1}{2} \sum_{j=1}^{d} f_j (\theta_j - \theta_j^0)^2. \quad (6)$$

Applying Eqs. (1) and (6) to Eq. (3), we obtain the EWC objective:

$$J_{\text{EWC}}(\boldsymbol{\theta}) = J_{\text{ERM}}(\boldsymbol{\theta}) + \frac{\lambda}{2} \sum_{j=1}^{d} f_j (\theta_j - \theta_j^0)^2, \quad (7)$$

where λ is the trade-off parameter.

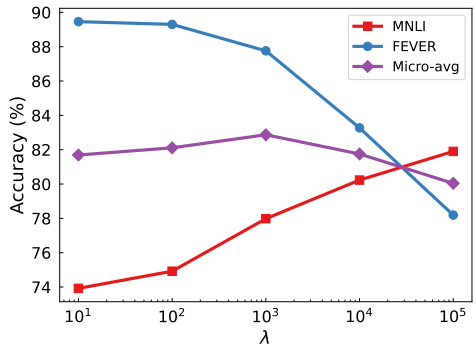
3 Proposed method

As shown in Figure 1, EWC requires extremely large values of λ to be effective. We analyze the components of EWC and find that this problem arises from the Fisher approximation in Eq. (5). The diagonal element f_j corresponds to the j^{th} element of the squared gradient with respect to $\boldsymbol{\theta}^0$. Since its training had already converged, the values of the gradient are typically small. When we square such a small decimal and combine it with the squared difference between the current and prior parameters, the final value can be vanishingly small.[2] We find that this issue is neither affected by datasets nor pre-trained language models. In Appendix A, we further investigate this issue on another pre-trained language model.

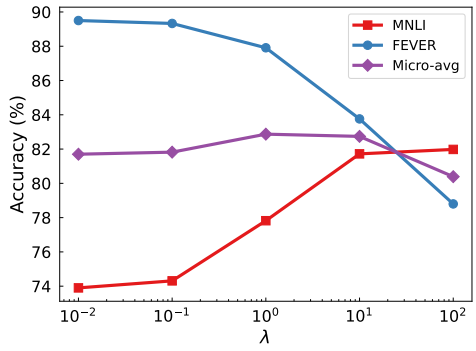
We propose scaling up the Fisher approximation by taking the square root to resolve the issue above. We define the square root of EWC (REWC) as:

$$J_{\text{REWC}}(\boldsymbol{\theta}) = J_{\text{ERM}}(\boldsymbol{\theta}) + \lambda\sqrt{A + \epsilon}, \quad (8)$$

[2]For example, in the MNLI⇒FEVER experiment, we find that 85.9% of non-zero f_j are less than 1e-10.



(a) REWC



(b) AEWC

Figure 2: Accuracy vs. trade-off parameter λ of our REWC and AEWC. Both methods begin to affect accuracy with much lower λ (i.e., 10^2 and 10^{-1} for REWC and AEWC, respectively) while maintaining average accuracies similar to EWC.

where $A = \sum_{j=1}^{d} f_j(\theta_j - \theta_j^0)^2$, and ϵ is a small value (e.g., 10^{-8}) for preventing the derivative of the square root at 0.

Another solution is to use the absolute value of the gradient instead of the squared gradient. We define:

$$\mathbf{g} = \frac{1}{N} \sum_{(x,y)\in\mathcal{S}^0} \left|\nabla_{\boldsymbol{\theta}^0} \mathcal{L}(h_{\boldsymbol{\theta}}(x), y)\right|. \quad (9)$$

Note that diag($\mathbf{g}$) is positive semi-definite (like diag($\mathbf{f}$)) because all of its eigenvalues are greater than or equal to 0. Replacing the squared difference with the absolute difference yields our absolute EWC (AEWC):

$$J_{\text{AEWC}}(\boldsymbol{\theta}) = J_{\text{ERM}}(\boldsymbol{\theta}) + \lambda \sum_{j=1}^{d} g_j |\theta_j - \theta_j^0|. \quad (10)$$

Figure 2 shows the results of REWC and AEWC based on the same setting as in Figure 1.

4 Experiments

4.1 Datasets

We evaluated the objective functions described in §2 and §3 on natural language inference and fact-checking tasks. We used six datasets pre-processed by Schuster et al. (2021) as follows:

MNLI (Williams et al., 2018) is a multi-genre natural language inference dataset. The task is to determine the inference relation between two sentences. Schuster et al. (2021) converted the original labels {"entailment", "contradiction","neutral"} into {"supported", "refuted", "not enough info"}.

FEVER (Thorne et al., 2018) (Fact Extraction and VERification) verifies whether a claim is supported or refuted by an evidence sentence, or decides whether there is insufficient information to make a decision.

VITC (Schuster et al., 2021) introduces the notion of contrastive evidence to FEVER. Given a claim, two evidence sentences that are nearly identical but with different labels are created. Thus, the task becomes more challenging than that of FEVER. The dataset contains both real and synthetic examples. We used only the real ones in our experiments.

ADVERSARIAL (Thorne et al., 2019) is derived from the FEVER 2.0 shared task, containing adversarially created claims that aim to induce erroneous predictions to the FEVER-trained models.

SYMMETRIC (Schuster et al., 2019) is another dataset that challenges the FEVER-trained models. It contains synthetically created claim-evidence pairs designed to break models that often make predictions using claims only without taking evidence sentences into account.

TRIGGERS (Atanasova et al., 2020) contains adversarial claims generated by using GPT-2 (Radford et al., 2019) given the original claims and triggers, which are words that cause the model to flip its prediction.

We selected λ that yields the highest average accuracy on the development (dev) sets. To avoid a bias towards more populated datasets (e.g., VITC), we created our balanced dev sets by randomly selecting 9,000 examples from each of the original dev sets. Since the dev and test sets of MNLI are identical, we split 9,000 examples from the training set to form the dev set and used the test set for the final evaluation. Table 1 shows our dataset statistics.

Dataset	|Train|	|Dev|	|Test|
MNLI	383,702	9,000	9,832
FEVER	178,059	9,000	11,710
VITC	248,953	9,000	34,481
ADVERSARIAL	–	–	766
SYMMETRIC	–	–	712
TRIGGERS	–	–	186

Table 1: Dataset statistics in our experiments. Bottom three datasets contain only test sets adversarially created for testing robustness of fact-checking models.

4.2 Training details

We implemented our base model described in §2.1 using Hugging Face's Transformers library (Wolf et al., 2020). Specifically, the model consists of a two-layer MLP and BERT-base. Let x be the input sequence (i.e., a pair of sentences in our datasets). BERT-base encodes x into a sequence of hidden state vectors. Following common practice, we used the first hidden state vector of the special **cl**assification token (i.e., [CLS]) to represent x and fed it to the MLP followed by a softmax function.

For all experiments, we used Adafactor optimizer (Shazeer and Stern, 2018) with a gradient clipping of 1.0. Our effective batch size is 256.[3] For standard training, we randomly initialized the model parameters with $\mathcal{N}(0, 0.02)$[4], except for those of BERT-base. We trained each model for three epochs with a learning rate of 2e-5.

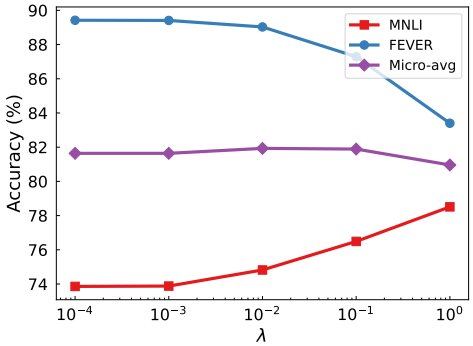
For sequential training, we randomly selected 1% of examples from $\mathcal{D}^0$ to represent $\mathcal{S}^0$ in Eq. (5). We also varied the subsample size from 0.1% to 10% but did not observe significant changes in performance. We initialized the current model parameters using the prior ones (i.e., $\boldsymbol{\theta}^0 \rightarrow \boldsymbol{\theta}$). Determining a learning rate can be challenging. We used a method analogous to the learning rate decay technique (Ng, 2017). Let α_0 be the initial learning rate and r be the number of prior training runs. We computed the learning rate α for the current training run as:

$$\alpha = \frac{1}{1 + (\text{decay\_rate} \times r)} \alpha_0. \qquad (11)$$

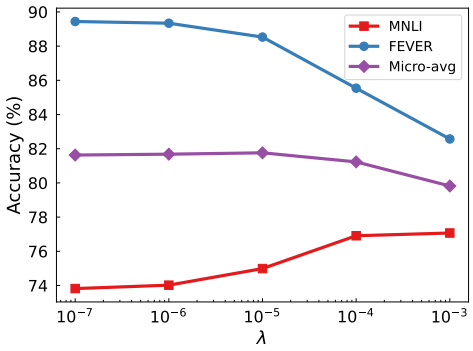
For example, consider the case of further training the MNLI-trained model on the FEVER dataset, where $\alpha_0 =$ 2e-5 and $r = 1$. We set decay_rate to 1e-2 for all sequential training experiments. Using Eq. (11), the learning rate α for the current run decreases to 1.98e-5. We conducted all the experiments on NVIDIA Tesla A100 GPUs.

[3]We used gradient accumulation with 8 batches of 32.

[4]This is the default setting in Transformers.



(a) EWC w/o f_j



(b) AEWC w/o g_j

Figure 3: Accuracy vs. trade-off parameter λ of EWC and AEWC without f_j and g_j, respectively.

4.3 Results

Table 2 shows the results of various settings on the test sets. For sequential training, conducting experiments on all combinations takes time and considerable resources. Thus, we chose only a representative order for the datasets in accordance with their publication times. Since the MNLI and FEVER datasets were published at the same time, we decided to start with MNLI due to its generality.

We considered the mix-and-retrain method ($\cup$) with ERM as the *topline* setting. Unsurprisingly, this method yields the best performance on the prior datasets. The sequential training method ($\Rightarrow$) with ERM (i.e., vanilla fine-tuning) encounters severe catastrophic forgetting on the prior datasets. Our REWC and AEWC effectively reduce the values of λ. AEWC requires the lowest λ among the three objective functions. The performances of all the methods seem comparable on average, but each yields a different trade-off in accuracy between the prior and current datasets. Regarding the training time, AEWC is faster than REWC/EWC (though not significant) because its computation is simpler.

Training set	Obj.	λ	MNLI	FEVER	VitC	Adver.	Sym.	Trig.
MNLI	ERM	–	83.9±0.1	67.7±0.7	47.8±0.7	51.0±0.8	74.8±0.3	68.3±1.4
FEVER	ERM	–	58.8±0.2	87.4±0.1	59.7±0.1	51.4±0.7	75.3±0.2	65.4±1.4
VitC	ERM	–	62.5±1.0	65.1±0.5	78.2±1.2	28.9±0.5	65.8±1.2	69.1±2.8
MNLI ∪ FEVER	ERM	–	83.9±0.2	87.8±0.1	61.0±0.3	53.8±0.2	82.6±0.4	73.8±0.4
MNLI ⇒ FEVER	ERM	–	74.9±0.2	88.2±0.2	62.7±0.1	55.0±0.3	82.6±0.2	71.2±0.4
	EWC	10^7	79.3±0.2	86.3±0.1	61.0±0.4	53.7±0.4	80.3±0.6	67.7±1.4
	REWC	10^3	78.7±0.2	86.8±0.1	61.5±0.3	53.6±0.5	81.1±0.6	69.2±0.6
	AEWC	10^0	78.7±0.2	87.2±0.1	61.9±0.3	53.9±0.4	81.3±0.4	70.5±0.4
FEVER ∪ VitC	ERM	–	69.0±0.4	87.5±0.1	83.3±0.3	51.0±0.2	79.0±0.7	71.5±0.8
FEVER ⇒ VitC	ERM	–	66.2±0.4	75.8±0.4	84.4±0.1	39.6±0.8	71.3±0.8	70.9±0.9
	EWC	10^6	65.8±0.3	78.2±0.2	83.6±0.1	40.5±1.4	71.3±0.5	70.0±0.6
	REWC	10^2	66.2±0.3	76.7±0.2	84.2±0.1	39.7±1.5	71.4±0.3	70.5±0.6
	AEWC	10^{-1}	66.3±0.4	76.3±0.2	84.3±0.1	39.5±1.4	71.4±0.4	70.6±0.6
MNLI ∪ VitC	ERM	–	84.0±0.1	76.8±0.2	84.3±0.1	43.8±0.6	75.5±0.6	74.6±1.9
MNLI ⇒ VitC	ERM	–	76.0±0.2	72.4±0.2	85.5±0.2	40.2±0.8	73.0±0.3	71.7±1.0
	EWC	10^5	76.5±0.3	72.7±0.4	85.3±0.1	41.0±1.0	73.3±0.5	72.4±1.8
	REWC	10^2	76.7±0.3	72.9±0.3	85.1±0.1	41.1±0.9	73.5±0.3	72.8±1.9
	AEWC	10^{-1}	76.4±0.2	72.7±0.4	85.3±0.1	40.7±1.1	73.3±0.4	72.8±1.9
MNLI ∪ FEVER ∪ VitC	ERM	–	83.8±0.2	88.1±0.1	84.6±0.1	53.5±0.6	82.6±0.4	73.2±1.0
MNLI ⇒ FEVER ⇒ VitC	ERM	–	75.1±0.3	79.1±0.3	85.7±0.0	44.4±0.5	75.4±0.7	74.9±0.7
	EWC	10^6	77.5±0.3	79.1±0.2	84.0±0.2	44.2±0.4	75.1±0.5	73.3±1.6
	REWC	10^2	76.4±0.4	77.7±0.2	85.2±0.1	42.9±0.4	74.4±0.5	73.5±0.9
	AEWC	10^0	78.6±0.2	82.8±0.1	80.0±1.0	46.8±0.6	76.9±0.5	74.1±1.5

Table 2: Symbol ∪ denotes mixing training sets, while arrow ⇒ denotes using training sets sequentially. Gray color highlights the effect of *catastrophic forgetting* on the prior dataset. Blue color emphasizes the performance on the current dataset. Green color indicates the topline performance of the mix-and-retrain method. We ran each experiment five times using different random seeds and report mean and standard deviation.

Obj.	λ	MNLI	FEVER	VitC
EWC	10^7	79.3±0.2	86.3±0.1	61.0±0.4
w/o f_j	10^{-2}	75.9±0.1	88.0±0.1	62.4±0.2
AEWC	10^0	78.7±0.2	87.2±0.1	61.9±0.3
w/o g_j	10^{-5}	76.1±0.2	87.6±0.1	62.1±0.2

Table 3: Ablation studies on EWC and AEWC for MNLI ⇒ FEVER. "w/o f_j (or g_j)" denotes omitting the gradient component from the regularization term.

4.4 Discussion

We can interpret the EWC family as a weighted sum of the squared (or absolute) differences between the current and prior parameters. The gradient component helps suggest which parameter is important. To examine the benefit of gradient information, we conducted ablation studies on EWC and AEWC in the MNLI ⇒ FEVER experiment. We omitted f_j and g_j from Eqs. (7) and (10), respectively. The remaining regularization terms resemble the squared ℓ_2-norm and the ℓ_1-norm that take the prior parameters into account.

As seen in Figure 3, without the gradient component, both methods need lower λ to affect the accuracy of the prior dataset (MNLI). However, improvements on the prior dataset are marginal (∼1%) before reaching the optimal average accuracy compared to the original EWC and AEWC (∼4%). Table 3 shows the ablation results on the test sets, indicating that omitting the gradient component yields lower accuracies on the prior dataset. These results confirm that the gradient component is indeed helpful.

5 Conclusion

Without realizing the diminishing effect of EWC, we may fine-tune a pre-trained language model with a conventional range of hyperparameters and find no effect in combating catastrophic forgetting. We identified a possible cause of this issue and suggested two alternative objective functions, REWC and AEWC, that yield results comparable to the original EWC. Exploring more efficient ways for choosing an optimal λ is part of our future work.

Acknowledgments

This work is supported by JST CREST Grants (JPMJCR18A6 and JPMJCR20D3) and MEXT KAKENHI Grants (21H04906), Japan.

References

Pepa Atanasova, Dustin Wright, and Isabelle Augenstein. 2020. Generating label cohesive and well-formed adversarial claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3168–3177, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xu He and Herbert Jaeger. 2018. Overcoming catastrophic interference using conceptor-aided backpropagation. In *6th International Conference on Learning Representations (ICLR)*.

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. 2017. Overcoming catastrophic forgetting in neural networks. *Proceeding of the National Academy of Science*, 114(13):3521–3526.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

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

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Andrew Ng. 2017. Learning rate decay. https://www.coursera.org/lecture/deep-neural-network/learning-rate-decay-hjgIA.

Razvan Pascanu and Yoshua Bengio. 2014. Revisiting natural gradient for deep networks. In *2nd International Conference on Learning Representations (ICLR)*.

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

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

Danielle Saunders, Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2019. Domain adaptive inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 222–228, Florence, Italy. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. 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 3419–3425, Hong Kong, China. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2062–2068, Minneapolis, Minnesota. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6, Hong Kong, China. Association for Computational Linguistics.

Vladimir Vapnik. 1992. Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann.

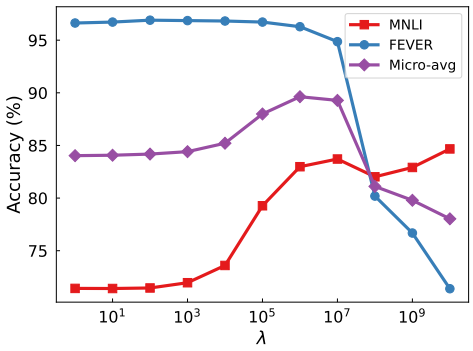
Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

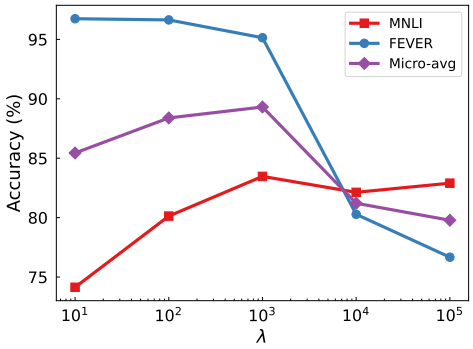
Jeffrey O. Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. 2020. Side-tuning: A baseline for network adaptation via additive side networks. In *Computer Vision – ECCV 2020*, pages 698–714, Cham. Springer International Publishing.

A Additional results

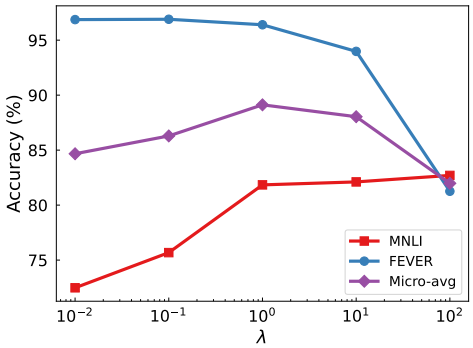
We verified the diminishing effect of EWC on another pre-trained language model, A Lite BERT (ALBERT, Lan et al. 2020). Figure 4 shows the results of sequential training: MNLI ⇒ FEVER. We can still see the diminishing effect of the original EWC, while our REWC and AEWC reduce the value of λ by three and six orders of magnitude and produced similar results.



(a) EWC



(b) REWC



(c) AEWC

Figure 4: Accuracy vs. trade-off parameter λ on the balanced dev sets of MNLI and FEVER using ALBERT-base (Lan et al., 2020). EWC, REWC, and AEWC achieve highest average accuracies with $\lambda = 10^6$, 10^3, and 10^0, respectively.