Elastic Weight Consolidation (EWC): Nuts and Bolts

Abhishek Aich
University of California, Riverside
aaich@ece.ucr.edu

Abstract

In this report, we present a theoretical support of the continual learning method **Elastic Weight Consolidation**, introduced in paper titled 'Overcoming catastrophic forgetting in neural networks' [4]. Being one of the most cited paper in regularized methods for continual learning, this report disentangles the underlying concept of the proposed objective function. We assume that the reader is aware of the basic terminologies of continual learning.

Keywords— Catastrophic forgetting, continual learning, EWC, neural networks

1 Introduction

Following are the notations used throughout this report. Vectors and matrices are denoted in bold lowercase and bold uppercase, respectively. Superscript $^\top$ denotes matrix transpose. $\mathbb{E}[\cdot]$ denotes the expectation operator. An optimum value of a variable is denoted by adding a superscript *.

Continual learning is a much desired attribute for neural networks. For example, if we train a model to distinguish between images of a cat and a dog (task 1), and subsequently train it again to distinguish between images of chair and table (task 2), the model should be able to retain its knowledge on task 1 even after learning task 2. In simple terms, our network model should be able to perform equally well on all seen tasks, even after learning new ones. Any degradation of performance on the previous tasks after learning new ones is fittingly termed as *catastrophic forgetting*. This sub-research area has seen an insurgence in works in recent times [1, 4, 5, 11]. Briefly, the continual learning scenarios can be categorized into following [9]:

- **Task-Incremental Learning**: For the given set of tasks, the task identity is known during testing.
- **Domain-Incremental Learning**: For the given set of tasks, task identity is not provided during testing, but need not infer the same.
- **Class-Incremental Learning**: For the given set of tasks, task identity is not provided during testing, but has to infer the same.

We highly recommend [9, 10] for a good overview of different methodologies to alleviate catastrophic forgetting as well as continual learning in general. The next section describes the well studied regularization method of continual learning: Elastic Weight Consolidation. It presents a solution to the continual learning problem by making task-specific synaptic (*read* network

parameters) consolidation. Based on the theory of plasticity of post-synaptic dendritic spines in the brain, this method presents a paradigm that marks how important is a network parameter to the previous tasks and penalizes any change made to it depending upon the importance, while learning new tasks.

2 Elastic Weight Consolidation

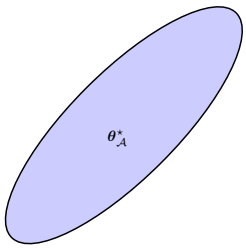
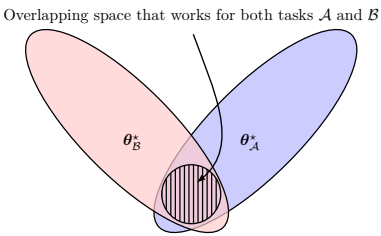


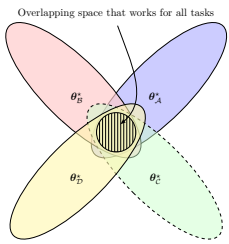
Figure 1: **Possible configurations of $\boldsymbol{\theta}^\star_{\mathcal{A}}$.** The shaded region represents a space of optimum $\boldsymbol{\theta}_{\mathcal{A}}$ with acceptable errors w.r.t. $\boldsymbol{\theta}^\star_{\mathcal{A}}$ for task $\mathcal{A}$.

Denote parameters of layers of a deep neural network (DNN) with $\boldsymbol{\theta}$. Training DNNs generates a mapping between the input distribution space and target distribution space. This is done by finding out an optimum $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$ which results in the least error in the training objective. It has been shown in earlier works [7] that such a mapping can be obtained with many configurations of $\boldsymbol{\theta}^\star$, represented in Fig. 1. The term *many configurations* can be interpreted as a solution space around the most optimum $\boldsymbol{\theta}$ with acceptable error in the learned mapping. Note that in figures to follow, the shaded ellipses represent the solution of individual tasks where as the overlapping region of multiple ellipses, marked by , represents the common solution space for all tasks.

Let's begin with a simple case of two tasks, task $\mathcal{A}$ and task $\mathcal{B}$. To have a configuration of parameters that performs well for both $\mathcal{A}$ and $\mathcal{B}$, the network should be able to pick $\boldsymbol{\theta}$ from the overlapping region of the individual solution spaces (see Fig. 2(a)). This is with the assumption that there is always an overlapping region for the solution spaces of all tasks for the network to learn them sequentially. A case of four tasks has been illustrated in Fig. 2(b). In the first instance, the network can learn any $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathcal{A}}$ that performs well for task $\mathcal{A}$. But with the arrival of task $\mathcal{B}$, the network should pick up a $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathcal{A},\mathcal{B}}$. The next question that arrives is how can the network learn the a set of parameters that lies in this overlapping region. To this end, EWC presents a method of selective regularization of parameters $\boldsymbol{\theta}$. After learning $\mathcal{A}$, this regularization method identifies which parameters are important for $\mathcal{A}$, and then penalizes the any change made to the network parameters according to their importance while learning $\mathcal{B}$.



(a) Example for two tasks



(b) Example for four tasks

Figure 2: **Overlap of possible configurations of $\boldsymbol{\theta}^\star$.** The overlapping space represents an optimum parameter region where the network performs without any catastrophic degradation on previous tasks.

To formulate the objective, we start by taking a Bayesian approach needed to estimate the

network parameters $\boldsymbol{\theta}$. More specifically given the data $\boldsymbol{\Sigma}$, we want to learn the posterior probability distribution function $p(\boldsymbol{\theta}|\boldsymbol{\Sigma})$. Following [2] and using Bayes rule, write

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{\Sigma})}_{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{\Sigma}|\boldsymbol{\theta})}^{\text{likelihood}} \overbrace{p(\boldsymbol{\theta})}^{\text{prior}}}{p(\boldsymbol{\Sigma})} \tag{1}$$

Since maximizing a function is same as maximizing its logarithm, we take $\log(\cdot)$ of (1) as follows.

$$\log(p(\boldsymbol{\theta}|\boldsymbol{\Sigma})) = \log(p(\boldsymbol{\Sigma}|\boldsymbol{\theta})) + \log(p(\boldsymbol{\theta})) - \log(p(\boldsymbol{\Sigma})) \tag{2}$$

To train the neural network on $\boldsymbol{\Sigma}$, the objective function to be optimized over the log-likelihood function.

$$\arg\max_{\boldsymbol{\theta}} \left\{ \ell(\boldsymbol{\theta}) = \log(p(\boldsymbol{\theta}|\boldsymbol{\Sigma})) \right\} \tag{3}$$

For the case of given two independent tasks such that $\boldsymbol{\Sigma} = \left\{\mathcal{A}, \mathcal{B}\right\}$ (with $\mathcal{B}$ appearing in sequence after $\mathcal{A}$), (2) can be written as

$$\begin{aligned} \log(p(\boldsymbol{\theta}|\boldsymbol{\Sigma})) &= \log(p(\mathcal{B}|\mathcal{A}, \boldsymbol{\theta})) + \log(p(\boldsymbol{\theta}|\mathcal{A})) - \log(p(\mathcal{B}|\mathcal{A})) \\ &= \log(p(\mathcal{B}|\boldsymbol{\theta})) + \log(p(\boldsymbol{\theta}|\mathcal{A})) - \log(p(\mathcal{B})) \quad (\because \mathcal{A} \text{ and } \mathcal{B} \text{ are independent}) \end{aligned} \tag{4}$$

Following (1), $p(\mathcal{B}|\boldsymbol{\theta})$ is the loss for current task $\mathcal{B}$, $p(\mathcal{B})$ is the likelihood for $\mathcal{B}$, and now posterior $p(\boldsymbol{\theta}|\mathcal{A})$ for $\mathcal{A}$ becomes prior for $\mathcal{B}$.

2.1 Intractability of posterior of $\mathcal{A}$ and it's approximation

Referring (4), it can be observed that we have to deal with the function $p(\boldsymbol{\theta}|\mathcal{A})$. This is the posterior function for $\mathcal{A}$ which contains the information about the parameters that explain $\mathcal{A}$ using the given network. As discussed in [4], this posterior function is said to be intractable. Basically, the intractability of $p(\boldsymbol{\theta}|\mathcal{A})$ can be interpreted as the function not existing in some interpretable form. Hence, it is difficult to estimate its quantiles. See [8] for an example.

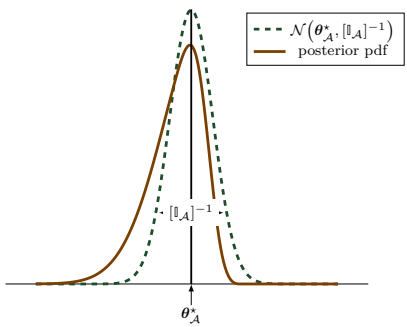


Figure 3: **Laplace approximation of true posterior pdf.** $\mathbb{I}_{\mathcal{A}}$ represents the Fisher Information matrix. See section 2.2 for more.

Next as the posterior is difficult to analyze in its present form, we aim to approximate it using Laplace approximation. In simple terms, Laplace approximation methodology is employed to find a normal distribution approximation to a continuous probability density distribution (see Fig. 3). Assuming $p(\boldsymbol{\theta}|\mathcal{A})$ is smooth and majorly peaked around its point of maxima (i.e. $\boldsymbol{\theta}^\star_{\mathcal{A}}$), we can approximate it with a normal distribution with mean $\boldsymbol{\theta}^\star_{\mathcal{A}}$ and variance $[\mathbb{I}_{\mathcal{A}}]^{-1}$. This brings us to the question on how did we come to the conclusion on these particular values of mean and variance for the normal distribution.

To begin with, compute the second order Taylor expansion of $\ell(\boldsymbol{\theta})$ around $\boldsymbol{\theta}^\star_{\mathcal{A}}$ as follows.

$$\ell(\boldsymbol{\theta}) \approx \ell(\boldsymbol{\theta}^\star_{\mathcal{A}}) + \left(\frac{\partial \ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^\star_{\mathcal{A}})^\top \left(\frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial^2 \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right)(\boldsymbol{\theta} - \boldsymbol{\theta}^\star_{\mathcal{A}}) + (\text{higher order terms}) \cdots \tag{5}$$

Neglecting higher order terms and noting that $\frac{\partial \ell(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}^\star_{\mathcal{A}}} = 0$ (slope of tangent at peak), we have

$$\ell(\boldsymbol{\theta}) \approx \ell(\boldsymbol{\theta}^\star_{\mathcal{A}}) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^\star_{\mathcal{A}})^\top \underbrace{\left(\frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial^2 \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right)}_{\text{Hessian}}(\boldsymbol{\theta} - \boldsymbol{\theta}^\star_{\mathcal{A}}) \tag{6}$$

Using (3), we can write (6) for task $\mathcal{A}$ as following.

$$\begin{aligned} \log(p(\boldsymbol{\theta}|\mathcal{A})) &= \log(p(\boldsymbol{\theta}^\star_{\mathcal{A}}|\mathcal{A})) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^\star_{\mathcal{A}})^\top \left(\frac{\partial^2 (\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial^2 \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right)(\boldsymbol{\theta} - \boldsymbol{\theta}^\star_{\mathcal{A}}) \\ \implies \log(p(\boldsymbol{\theta}|\mathcal{A})) &= \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^\star_{\mathcal{A}})^\top \left(\frac{\partial^2 (\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial^2 \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right)(\boldsymbol{\theta} - \boldsymbol{\theta}^\star_{\mathcal{A}}) + \Delta \end{aligned} \tag{7}$$

where $\Delta = \log\Big(p\Big(\boldsymbol{\theta}^\star_{\mathcal{A}}|\mathcal{A}\Big)\Big)$.

Next, write $\left(\frac{\partial^2 (\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial^2 \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right)$ as $-\left(\left(-\frac{\partial^2 (\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial^2 \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right)^{-1}\right)^{-1}$ and replace it back in (7) to express the same in the standard form of normal distribution function.

$$p(\boldsymbol{\theta}|\mathcal{A}) = \epsilon \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^\star_{\mathcal{A}})^\top \left(\left(-\frac{\partial^2 (\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial^2 \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right)^{-1}\right)^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}^\star_{\mathcal{A}})\right) \tag{8}$$

where $\epsilon = \exp(\Delta)$ is a constant. From (8), it can be concluded that we have obtained the Laplace approximation of posterior pdf as

$$p(\boldsymbol{\theta}|\mathcal{A}) \sim \mathcal{N}\left(\boldsymbol{\theta}^\star_{\mathcal{A}}, \left(-\frac{\partial^2 (\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial^2 \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right)^{-1}\right)$$

This has been illustrated in Fig. 3.

2.2 Importance of parameters using Fisher Information matrix

Notice the variance of the estimated normal distribution of $p(\boldsymbol{\theta}|\mathcal{A})$. Given $\boldsymbol{\theta}^\star_{\mathcal{A}}$, the term $\log(p(\boldsymbol{\theta}|\mathcal{A}))$ represents the log-likelihood of posterior pdf $p(\boldsymbol{\theta}|\mathcal{A})$. Clearly, the term represents the inverse of **Fisher information matrix** (FIM), $\mathbb{I}_{\mathcal{A}} = \mathbb{E}\left[-\frac{\partial^2 (\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial^2 \boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right]$. More

on this in **Appendix A.2** of [9]. Finally, we get $p(\boldsymbol{\theta}|\mathcal{A}) \sim \mathcal{N}\big(\boldsymbol{\theta}^\star_{\mathcal{A}}, [\mathbb{I}_{\mathcal{A}}]^{-1}\big)$. Further, as FIM can also be computed from first order derivatives, we can avoid the Hessian computed in (6) using the following property [3].

$$\begin{aligned}\mathbb{I}_{\mathcal{A}} &= \mathbb{E}\left[-\frac{\partial^2(\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial^2\boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right] \\ &= \mathbb{E}\left[\left(\left(\frac{\partial(\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial\boldsymbol{\theta}}\right)\left(\frac{\partial(\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial\boldsymbol{\theta}}\right)^\top\right)\bigg|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right] \end{aligned} \tag{9}$$

Putting (7) back together in (4), we have

$$\begin{aligned}\log(p(\boldsymbol{\theta}|\boldsymbol{\Sigma})) &= \log(p(\mathcal{B}|\boldsymbol{\theta})) + \log(p(\boldsymbol{\theta}|\mathcal{A})) - \log(p(\mathcal{B})) \\ &= \log(p(\mathcal{B}|\boldsymbol{\theta})) + \frac{\lambda}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^\star_{\mathcal{A}})^\top\left(\frac{\partial^2(\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial^2\boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right)(\boldsymbol{\theta}-\boldsymbol{\theta}^\star_{\mathcal{A}}) + \epsilon' \end{aligned} \tag{10}$$

where ϵ' accounts for all constants and λ is a hyper-parameter introduced to have a trade off between learning $\mathcal{B}$ and not forgetting $\mathcal{A}$. Simplifying more, we have

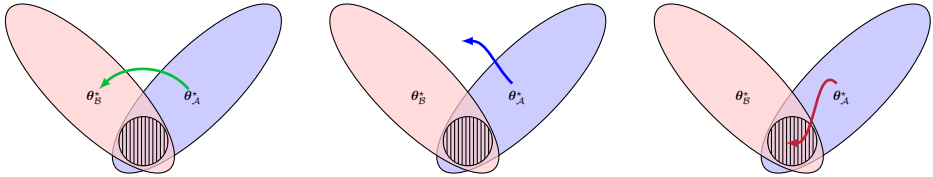
$$\begin{aligned}\log(p(\boldsymbol{\theta}|\boldsymbol{\Sigma})) &= \log(p(\mathcal{B}|\boldsymbol{\theta})) + \frac{\lambda}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^\star_{\mathcal{A}})^\top\left(\frac{\partial^2(\log(p(\boldsymbol{\theta}|\mathcal{A})))}{\partial^2\boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^\star_{\mathcal{A}}}\right)(\boldsymbol{\theta}-\boldsymbol{\theta}^\star_{\mathcal{A}}) + \epsilon' \\ &= \log(p(\mathcal{B}|\boldsymbol{\theta})) - \frac{\lambda}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^\star_{\mathcal{A}})^\top\mathbb{I}_{\mathcal{A}}(\boldsymbol{\theta}-\boldsymbol{\theta}^\star_{\mathcal{A}}) + \epsilon' \qquad \text{(Using (9))} \\ \implies \underbrace{\ell(\boldsymbol{\theta})}_{\text{overall loss}} &= \underbrace{\ell_{\mathcal{B}}(\boldsymbol{\theta})}_{\text{loss for }\mathcal{B}} - \underbrace{\frac{\lambda}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}^\star_{\mathcal{A}})^\top\mathbb{I}_{\mathcal{A}}(\boldsymbol{\theta}-\boldsymbol{\theta}^\star_{\mathcal{A}})}_{\text{weight regularizer}} + \epsilon' \end{aligned} \tag{11}$$

Further simplification of (11) can be found in [9]. Before we end this section, let's discuss how does the FIM indicates the **importance** of the parameters for the previous tasks.

We say a network has learnt a task when its objective has reached a minimum in the loss surface. We know that the curvature of such surfaces represent the sensitivity of the network with respect to the optimum $\boldsymbol{\theta}^\star$. This sensitivity can be determined by looking at the direction along which $\boldsymbol{\theta}^\star$ changes. This implies the curvature is inversely proportional to change in $\boldsymbol{\theta}^\star$. Hence, if the more the curvature, a 'δ' increment can result in large increase in the loss. Curvature of a curve is denoted by its Hessian and hence in our case, as the second derivative is of the log likelihood function of the posterior pdf, the FIM $\mathbb{I}_{\mathcal{A}}$ comes into picture. Thus, $\mathbb{I}_{\mathcal{A}}$ can tell us which parameter is important to the the previous task as its corresponding element in $\mathbb{I}_{\mathcal{A}}$ will have a large value, indicating higher importance. See [6] for more.

3 Conclusion

The EWC methodology alleviates catastrophic forgetting by regularizing parameters of a network trained on previous tasks by penalizing any change in them according to their importance. This importance is indicated by the Fisher information matrix i.e. after a network is trained on one task, fine-tuning on the next task is performed on according to (11). In the end, we refer to Fig. 4 for a pictorial representation of EWC. Fig. 4(a) represents the case when we simply fine-tune the network on the subsequent tasks. This makes the network learn the optimum parameters according to the current task, resulting in forgetting. Fig. 4(b) represents the case when we apply a stringent regularization to the parameters learnt from the previous



(a) No regularization (b) Stringent regularization (c) Flexible regularization

Figure 4: **Sequential training on task $\mathcal{B}$ after task $\mathcal{A}$.** 4(a): Train the network as it is: results in 'Forgetting', 4(b): Make no change in the parameters of previous tasks, 4(c): Make changes in the parameters of the previous tasks depending on their importance

tasks. This method doesn't compute the importance of the parameters and penalizes all of them equally. This may result in the network not only forgetting the previous task, but also not be able to learn the current one. Finally, Fig. 4(b) refers to the EWC methodology of computing the importance of parameters before fine-tuning on new tasks. This ensures that the network learns the optimum parameters that performs well for all tasks and hence, lies in the overlapping region of the solution spaces of the tasks in the given sequence.

REFERENCES

[1] Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 139–154.

[2] Herranz, L. Rotating networks to prevent catastrophic forgetting (www.lherranz.org/2018/08/21/rotating-networks-to-prevent-catastrophic-forgetting/), 2018.

[3] Kay, S. M. *Fundamentals of statistical signal processing.* Prentice Hall PTR, 1993.

[4] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences 114*, 13 (2017), 3521–3526.

[5] Li, Z., and Hoiem, D. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence 40*, 12 (2017), 2935–2947.

[6] Maltoni, D., and Lomonaco, V. Continuous learning in single-incremental-task scenarios. *Neural Networks 116* (2019), 56–73.

[7] Sussmann, H. J. Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural networks 5*, 4 (1992), 589–593.

[8] Tokdar, S. T. Lecture notes in STA 250: Statistics (Notes 11), Fall 2013.

[9] van de Ven, G. M., and Tolias, A. S. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734* (2019).

[10] Wiewel, F., and Yang, B. Localizing catastrophic forgetting in neural networks. *arXiv preprint arXiv:1906.02568* (2019).

[11] Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), JMLR. org, pp. 3987–3995.