Regularizing Trajectories to Mitigate Catastrophic Forgetting

Anonymous authors
Paper under double-blind review

Abstract

Regularization-based continual learning approaches generally prevent catastrophic forgetting by augmenting the training loss with an auxiliary objective. However in practical optimization scenarios with noisy data and/or gradients, it is possible that stochastic gradient descent can inadvertently change critical parameters. In this paper, we argue for the importance of regularizing optimization trajectories directly. We derive a new *co-natural* gradient update rule for continual learning whereby the new task gradients are preconditioned with the empirical Fisher information of previously learnt tasks. We show that using the co-natural gradient systematically reduces forgetting in continual learning. Moreover, it helps combat overfitting when learning a new task in a low-resource scenario.[1]

1 Introduction

> It is good to have an end to journey toward;
> but it is the journey that matters, in the end.
>
> *Ursula K. Le Guin*

Endowing machine learning models with the capability to learn a variety of tasks in a sequential manner is critical to obtain agents that are both versatile and persistent. However, continual learning of multiple tasks is hampered by catastrophic forgetting (McCloskey & Cohen, 1989; Ratcliff, 1990), the tendency of previously acquired knowledge to be overwritten when learning a new task.

Techniques to mitigate catastrophic forgetting can be roughly categorized into 3 lines of work (see Parisi et al. (2019) for a comprehensive overview): 1. regularization-based approaches, where forgetting is mitigated by the addition of a penalty term in the learning objective (Kirkpatrick et al. (2017); Chaudhry et al. (2018a), *inter alia*), 2. dynamic architectures approaches, which incrementally increase the model's capacity to accomodate the new tasks (Rusu et al., 2016), and 3. memory-based approaches, which retain data from learned tasks for later reuse (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2018b; 2019). Among these, regularization-based approaches are particularly appealing because they do not increase the model size and do not require access to past data. This is particularly relevant to real-world scenarios where keeping data from previous training tasks may be impractical because of infrastructural or privacy-related reasons. Moreover, they are of independent intellectual interest because of their biological inspiration rooted in the idea of synaptic consolidation (Kirkpatrick et al., 2017).

A good regularizer ensures that, when learning a new task, gradient descent will ultimately converge to parameters that yield good results on the new task while preserving performance on previously learned tasks. Critically, this is predicated upon successful optimization of the regularized objective, a fact that has been largely taken for granted in previous work. Non-convexity of the loss function, along with noise in the data (due to small or biased datasets) or in the gradients (due to stochastic gradient descent), can yield optimization trajectories — and ultimately convergence points — that are highly non-deterministic, even for the same starting parameters. As we demonstrate in this paper, this can cause unintended catastrophic forgetting along the optimization path. This is illustrated in a toy setting in Figure 1: a two parameter model is trained to perform task T_2 (an arbitrary bi-modal loss function) after having learned task T_1 (a logistic regression task). Standard finetuning, even in

[1] We commit to releasing the code to implement our method and reproduce our experiments upon acceptance.

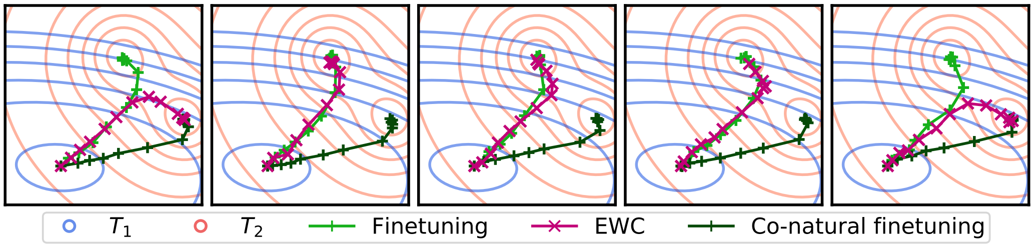


Figure 1: **On the importance of trajectories: an example with 2-dimensional logistic regression.** Having learned task T_1, the model is trained on T_2 with two different objectives: minimizing the loss on T_2 (Finetuning) and a regularized objective (EWC; Kirkpatrick et al. (2017)). We add a small amount of Gaussian noise to gradients in order to simulate the stochasticity of the trajectory. Plain finetuning and EWC often converge to a solution with high loss for T_1, but the co-natural optimization trajectory *consistently* converges towards the optimum with lowest loss for T_1.

the presence of a regularized objective (EWC; Kirkpatrick et al. (2017)), quickly changes the loss of T_1 and tends converge to a solution with high T_1 loss.

We propose to remedy this issue by regularizing the optimization trajectory itself, specifically by preconditioning gradient descent with the empirical Fisher information of previously learned tasks (§3). This yields what we refer to as a *co-natural* gradient, an update rule inspired by the natural gradient (Amari, 1997), but taking the Fisher information of *previous tasks* as a natural Riemannian metric[2] of the parameter space, instead of the Fisher information of the task being optimized for. When we introduce our proposed co-natural gradient for the toy example of Figure 1, the learning trajectory follows a path that changes the loss on T_1 much more slowly, and tends to converges to the optimum that incurs the lowest performance degradation on T_1.

We test the validity of our approach in a continual learning scenario (§4). We show that the co-natural gradient consistently reduces forgetting in a variety of existing continual learning approaches by a factor of ≈ 1.5 to 9, and greatly improves performance over simple finetuning, without modification to the training objective. We further investigate the special case of transfer learning in a two-task, low-resource scenario. In this specific case, control over the optimization trajectory is particularly useful because the optimizer has to rely on early stopping to prevent overfitting to the meager amount of training data in the target task. We show that the co-natural gradient yields the best trade-offs between source and target domain performance over a variety of hyper-parameters (§5).

2 Background and Notations

We first give a brief overview of the continual learning paradigm and existing approaches for overcoming catastrophic forgetting.

2.1 Notation

Let us define a task as a triplet containing an input space $\mathcal{X}$ and an output space $\mathcal{Y}$, both measurable spaces, as well as a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. In general, learning a task will consist of training a model to approximate the conditional distribution $p(y \mid x)$ induced by $\mathcal{D}$.

Consider a probabilistic model p_θ parametrized by $\theta \in \mathbb{R}^d$ where d is the size of the model, trained to perform a *source* task $S = \langle \mathcal{X}_S, \mathcal{Y}_S, \mathcal{D}_S \rangle$ to some level of performance, yielding parameters θ_S. In the most simple instance of continual learning, we are tasked with learning a second *target* task $T = \langle \mathcal{X}_T, \mathcal{Y}_T, \mathcal{D}_T \rangle$. In general in a multitask setting, it is not the case that the input or output spaces are the same. The discrepancy between input/output space can be addressed in various ways, *e.g.* by adding a minimal number of task-specific parameters (for example, different softmax layers for different label sets). To simplify exposition, we set these more specific considerations aside for now, and assume that $\mathcal{X}_S = \mathcal{X}_T$ and $\mathcal{Y}_S = \mathcal{Y}_T$.

[2] Informally, the reader can think of a Riemannian metric as a function that assigns an inner product $u, v \mapsto g_{\mathbf{x}}(u, v)$ to each point x in the space, thus inducing a localized notion of distance and curvature.

At any given point during training for task T, our objective will be to minimize the loss function $\mathcal{L}_T(\theta)$ – generally the expected log-likelihood $\mathbb{E}_{x,y\sim\mathcal{D}_T}[-\log p_\theta(y \mid x)]$. Typically, this will be performed by iteratively adding incremental update vectors $\delta \in \mathbb{R}^d$ to the parameters $\theta \longleftarrow \theta + \delta$.

2.2 Existing Approaches for Continual Learning

In this paper, we focus on those models that have a fixed architecture over the course of continual learning. The study of continual learning for models of fixed capacity can be split into two distinct (but often overlapping) streams of work:

Regularization-based approaches introduce a penalty in the loss function $\mathcal{L}_T$, typically quadratic, pushing the weights θ back towards θ_S:

$$\mathcal{L}_T(\theta) = \underbrace{\mathbb{E}_{x,y\sim\mathcal{D}_T} - \log p_\theta(y \mid x)}_{\text{NLL on task } T} + \underbrace{\lambda(\theta - \theta_S)^T \Omega_S (\theta - \theta_S)}_{\text{Regularization term}} \tag{1}$$

where Ω_S is a matrix, typically diagonal, that encodes the respective importance of each parameter with respect to task S, and λ is a regularization strength hyper-parameter. Various choices have been proposed for Ω_S; the diagonal empirical Fisher information matrix (Kirkpatrick et al., 2017), or path-integral based importance measures (Zenke et al., 2017; Chaudhry et al., 2018a). More elaborate regularizers have been proposed based on *e.g.* a Bayesian formulation of continual learning (Nguyen et al., 2017; Ahn et al., 2019) or a distillation term (Li & Hoiem, 2016). The main advantage of these approaches is that they do not rely on having access to training data of previous tasks.

Memory-based approaches store data from previously seen tasks for re-use in continued learning, either as a form of constraint, by *e.g.* ensuring that training on the new task doesn't increase the loss on previous tasks (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2018b), or for replay *i.e.* by retraining on instances from previous tasks (Rebuffi et al., 2017; Chaudhry et al., 2019; Aljundi et al., 2019b;a). Various techniques have been proposed for the selection of samples to store in the memory (Chaudhry et al., 2019; Aljundi et al., 2019b) or for retrieval of the samples to be used for replay Aljundi et al. (2019a).

All of these methods rely on stochastic gradient descent to optimize their regularized objective or to perform experience replay, with the notable exception of GEM (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2018b), where the gradients are projected onto the orthogonal complement of previous task's gradients. However, this method has been shown to perform poorly in comparison with simple replay (Chaudhry et al., 2019), and it still necessitates access to data from previous tasks.

3 Regularizing the Trajectory

After briefly recalling how the usual update is obtained in gradient descent, we derive a new, *co-natural* update designed to better preserve the distribution induced by the model over previous tasks.

3.1 Warm up: the Standard Gradient Descent Update

At point θ in the parameter space, gradient descent finds the optimal update δ that is (1) small and (2) locally minimizes the decrease in loss $\mathcal{L}(\theta + \delta) - \mathcal{L}(\theta)$ ($\approx \delta^\intercal \nabla_\theta \mathcal{L}$ at the first order). Traditionally this can be formulated as minimizing the Lagrangian:

$$\mathbb{L}(\delta) = \underbrace{\delta^\intercal \nabla_\theta \mathcal{L}_T}_{\substack{\text{first order} \\ \text{loss minimization term}}} + \underbrace{\mu \|\delta\|^2}_{\text{“small update” term}} \tag{2}$$

with Lagrangian multiplier $\mu > 0$. Minimizing $\mathbb{L}$ for δ yields the well-known optimal update δ^*:

$$\delta^* = -\frac{1}{2\mu} \nabla_\theta \mathcal{L}_T \tag{3}$$

where $\frac{1}{2\mu}$ corresponds to the learning rate (see Appendix A.1 for the full derivation).

3.2 KL Regularization of Trajectories

The $\|\delta\|^2$ term in $\mathbb{L}$ implicitly expresses the underlying assumption that the best measure of distance between parameters θ and $\theta + \delta$ is the Euclidean distance. In a continual learning setting however, the quantity we are most interested in preserving is the probability distribution that θ models on the source task S:

$$p_\theta^S(x, y) = p_\theta(y \mid x)p^S(x) \tag{4}$$

Therefore, a more *natural* distance between θ and $\theta + \delta$ is the Kullback-Leibler divergence $\text{KL}(p_\theta^S \| p_{\theta+\delta}^S)$ (Kullback & Leibler, 1951). For preventing catastrophic forgetting along the optimization path, we incorporate incorporate this KL term into the Lagrangian $\mathbb{L}$ itself:

$$\mathbb{L}(\delta) = \delta^\intercal \nabla_\theta \mathcal{L}_T + \mu\|\delta\|^2 + \nu \text{KL}(p_\theta^S \| p_{\theta+\delta}^S) \tag{5}$$

Doing so means that the optimization trajectory will tend to follow the direction that changes the distribution of the model the least. Notably, this is not a function of the previous objective $\mathcal{L}_S$, so knowledge of the original training objective is not necessary during continual learning (which is typically the case in path-integral based regularization methods (Zenke et al., 2017) or experience replay (Chaudhry et al., 2019)).

3.3 Co-natural Gradient Optimization

Presuming that δ is small, we can perform a second order Taylor approximation of the function $\delta \mapsto \text{KL}(p_\theta^S \| p_{\theta+\delta}^S)$ around 0. Considering that both the zeroeth and first order terms are null because 0 is a global minimizer of $\delta \mapsto \text{KL}(p_\theta^S \| p_{\theta+\delta}^S)$, this reduces the Lagrangian to a quadratic optimization problem (we refer the reader to Pascanu & Bengio (2013) for a more detailed derivation.):

$$\mathbb{L}(\delta) = \delta^\intercal \nabla_\theta \mathcal{L}_T + \mu\|\delta\|^2 + \frac{1}{2}\nu \delta^\intercal F_\theta^S \delta \tag{6}$$

where F_θ^S is the Hessian of the KL divergence around θ. A crucial, well-known property of this matrix is that it coincides with the Fisher information matrix[3] $\mathbb{E}_{x,y\sim p_\theta}[(\nabla \log p_\theta^S)(\nabla \log p_\theta^S)^T]$ (the expectation being taken over the model's distribution p_θ; see Appendix A.1 for details). This is appealing from a computational perspective because the Fisher can be computed by means of first order derivatives only.

Minimizing for δ yields the following optimal update:

$$\delta^* = -\lambda \left[F_\theta^S + \alpha I\right]^{-1} \nabla_\theta \mathcal{L}_T \tag{7}$$

where coefficients μ and ν are folded into two hyper-parameters: the learning rate λ and a damping coefficient α (the step-by-step derivation can be found in Appendix A.1). In practice, especially with low damping coefficients, it is common to obtain updates that are too large (typically when some parameters have no effect on the KL divergence). To address this, we re-normalize δ^* to have the same norm as the original gradient, $\|\nabla \mathcal{L}_T\|$.

For computational reasons, we will make 3 key practical approximations to the Fisher:

1. $F_\theta^S \approx F_{\theta_S}^S$: we maintain the Fisher computed at θ_S, instead of recomputing F_S at every step of training. This relieves us of the computational burden of updating the Fisher for every new value of θ.
2. F^S is diagonal: this is a common approximation in practice with two appealing properties. First, this makes it possible to store the d diagonal Fisher coefficients in memory. Second, this trivializes the inverse operation (simply invert the diagonal elements).

[3] Hence our use of the letter F to designate the Hessian

3. Empirical Fisher: this common approximation replaces the expectation under the model's distribution by the expected log-likelihood of the *true* distribution: $E_{x,y\sim p^S}[(\nabla \log p^S_\theta)(\nabla \log p^S_\theta)^T]$ (mind the subscript). This is particularly useful in tasks with a large or unbounded number of classes (*e.g.* structured prediction), where summing over all possible outputs is intractable. We can then compute the diagonal of the empirical Fisher using Monte Carlo sampling: $\frac{1}{N}\sum_{i=1}^{N}[\nabla \log p^S_\theta(y_i \mid x_i)]^2$ with (x_i, y_i) sampled from $\mathcal{D}_S$ (we use $N = 1000$ for all experiments).

This formulation bears many similarities with the natural gradient from Amari (1997), which also uses the KL divergence as a metric for choosing the optimal update δ^*. There is a however a crucial difference, both in execution and purpose: where the natural gradient uses knowledge of the curvature of the KL divergence of $\mathcal{D}_T$ to *speed-up* convergence, our proposed method leverages the curvature of the KL divergence on $\mathcal{D}_S$ to *slow-down* divergence from $p^S_{\theta_S}$. To highlight the resemblance and complementarity between these two concepts, we refer to the new update as the *co-natural* gradient.

3.4 Beyond Two Tasks

In a continual learning scenario, we are confronted with a large number of tasks $T_1 \dots T_n$ presented in sequential order. When learning T_n, we can change the Lagrangian $\mathbb{L}$ from 5 to incorporate the constraints for all previous tasks $T_1 \dots T_{n-1}$:

$$\mathbb{L}(\delta) = \delta^\intercal \nabla_\theta \mathcal{L}_{T_n} + \mu\|\delta\|^2 + \sum_{i=1}^{n-1} \nu_i \mathrm{KL}(p^{T_i}_\theta \| p^{T_i}_{\theta+\delta}) \tag{8}$$

This in turn changes the Fisher in Eq. 8 to $\tilde{F}_{n-1} := \frac{1}{2}\sum_{i=1}^{n-1}\nu_i F^{T_i}$. The choice of the coefficients ν_i is crucial. Setting all ν_i to the same value, *i.e.* assigning the same importance to all tasks is suboptimal for a few reasons. First and foremost, it is unreasonable to expect of a model with finite capacity to remember an unbounded number of tasks (as tasks "fill-up" the model capacity, $\tilde{F}_{n-1}$ is likely to become more "homogeneous"). Second, as training progresses and θ changes, our approximation that $F^{T_i}_\theta \approx F^{T_i}_{\theta_{T_i}}$ is less and less likely to hold.

We address this issue in the same fashion as Schwarz et al. (2018), by keeping a rolling exponential average of the Fisher matrices:

$$\tilde{F}^\gamma_n = \gamma F_{T_n} + (1-\gamma)\tilde{F}^\gamma_{n-1} \tag{9}$$

In this case, previous tasks are gracefully forgotten at an exponential rate controlled by γ. We account for the damping α term in Eq. 7 by setting $\tilde{F}_0 := \frac{\alpha}{\gamma}I$. In preliminary experiments, we have found $\gamma = 0.9$ to yield consistently good results, and use this value in all presented experiments.

4 Continual Learning Experiments

4.1 Experimental setting

To corroborate our hypothesis that controlling the optimization trajectory with the co-natural gradient reduces catastrophic forgetting, we perform experiments on two continual learning testbeds:

- **Split CIFAR**: The CIFAR100 dataset, split into 20 independent 5-way classification tasks. Similarly to Chaudhry et al. (2018b), we use a smaller version of the ResNet architecture (He et al., 2016).
- **Omniglot**: the Omniglot dataset (Lake et al., 2015) consists of 50 independent character recognition datasets on different alphabet. We adopt the setting of Schwarz et al. (2018) and consider each alphabet as a separate task.[4] On this dataset we use the same small CNN architecture as Schwarz et al. (2018).

[4] Note that this is a different setting than the usual meta-learning scenario that Omniglot is used for.

Table 1: Average accuracies and forgetting after all tasks have been learnt, with and without the co-natural gradient. Results are reported in percentages (± the standard deviation over 5 re-runs). Bold print indicates statistically significant difference between standard and co-natural ($p < 0.05$).

	Split CIFAR			Omniglot		
	Finetuning	EWC	ER	Finetuning	EWC	ER
	Average accuracy ↑					
Standard	35.92 ±1.19	44.86 ±2.01	61.08 ±0.94	16.31 ±1.05	70.31 ±3.46	70.90 ±0.80
Co-natural	**56.82** ±1.47	**56.50** ±1.28	59.34 ±2.02	**71.25** ±4.90	69.11 ±4.70	**75.48** ±1.92
	Forgetting ↓					
Standard	34.05 ±0.99	17.50 ±2.09	10.66 ±0.70	77.26 ±1.11	14.16 ±2.35	22.43 ±0.80
Co-natural	**5.53** ±1.12	**4.91** ±0.92	**5.25** ±1.20	**8.49** ±2.75	**8.73** ±1.81	**5.21** ±0.30

We adopt the experimental setup from Chaudhry et al. (2019): in each dataset we create a "validation set" of 3 tasks, used to select the best hyper-parameters, and keep the remaining tasks for evaluation. This split is chosen at random and kept the same across all experiments. In these datasets, the nature and possibly the number of classes for each task changes. We account for this by training a separate softmax layer for each task, and apply continual learning only to the remaining, "feature-extraction" part of the model.

We report results along two common metrics for continual learning: **average accuracy**, the accuracy at the end of training averaged over all tasks, and **forgetting**. Forgetting is defined in Chaudhry et al. (2018a) as the difference in performance on a task between the current model and the best performing model on this task. Formally if A_t^T represents the accuracy on task T at step t of training, the forgetting F_t^T at step t is defined as $F_t^T = \max_{\tau<t} A_\tau^T - A_t^T$. 'Low forgetting' means that the model tend to keep the same level of performance on a task it has learned.

We implement the co-natural update rule on top of 3 baselines:

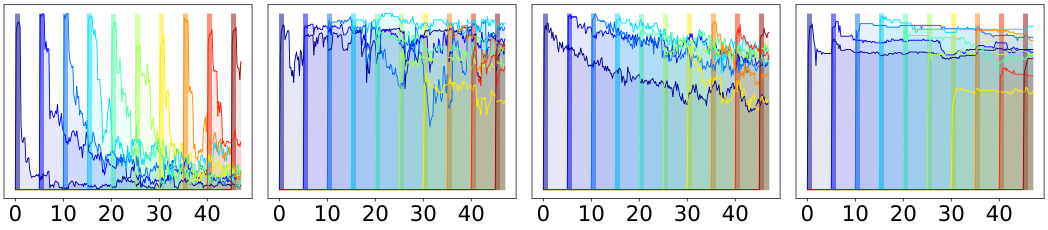
- **Finetuning**: Simply train the model on the task at hand, without any form of regularization.
- **EWC**: Proposed by Kirkpatrick et al. (2017), it is a simple but effective quadratic regularization approach. While neither the most recent nor sophisticate regularization technique, it is a natural baseline for us to compare to in that it also consists in a Fisher-based penalty — although in the loss function instead of the optimization dynamics.
- **ER**: Experience replay with a fixed sized episodic memory proposed by Chaudhry et al. (2019). While not directly comparable to EWC in that it presupposes access to data from previous tasks, ER is a simple approach that boasts the best performances on a variety of benchmarks (Chaudhry et al., 2019). In all experiments, we use memory size 1,000 with reservoir sampling.

Training proceeds as follows: we perform exhaustive search on all the hyper-parameter combinations using the validation tasks. Every combination is reran 3 times (the order of tasks, model initialization and order of training examples changes with each restart), and rated by accuracy averaged over tasks and restarts. We then evaluate the best hyper-parameters by continual training on the evaluation tasks. Results are reported over 5 random restarts, and we control for statistical significance using a paired t-test (we pair together runs with the same task ordering). We refer to Appendix A.2 for more details regarding fine-grained design choices.

4.2 Results

The upper half of Table 1 reports the average accuracy of all the tasks at the end of training (higher is better). We observe that the co-natural gradient always improves greatly over simple finetuning, and occasionally over EWC and ER. We note that on both datasets, bare-bone co-natural finetuning matches or exceeds the performance of EWC and ER even though it requires strictly fewer resources (no need to store the previous parameters as in EWC, or data in ER).

Even more appreciable is the effect of the co-natural trajectories on forgetting, as shown in the lower half of Table 1. As evidenced by the results in the lowest row, using the co-natural gradient systematically results in large drops in forgetting across all approaches and both datasets, even when the average accuracy is not increased.



(a) Finetuning (b) EWC (c) ER (d) Co-natural Finetuning

Figure 2: Evolution of task performance over the course of continual learning on one ordering of Omniglot. For visibility we only show accuracies for every fifth task. The rectangular shaded regions delineate the period during which each task is being trained upon; with the exception of ER, this is the only period the model has access to the data for this task.

To get a qualitative assessment of the learning trajectories that yield such results, we visualize the accuracy curves of 10 out of the 47 evaluation tasks of Omniglot in Figure 2. We observe that previous approaches do poorly at keeping stable levels of performance over a long period of time (especially for tasks learned early in training), a problem that is largely resolved by the co-natural preconditioning. This seems to come at the cost of more intransigence (Chaudhry et al., 2018a), *i.e.* some of the later tasks are not being learnt properly. In models of fixed capacity, there is a natural trade-off between intransigence and forgetting (see also the "stability-plasticity" dilemma in neuroscience Grossberg (1982)). Our results position the co-natural gradient as a strong low-forgetting/moderate intransigence basis for future work.

5 Low-Resource Adaptation Experiments

In this section we take a closer look at the specific case of adapting a model from a single task to another, when we only have access to a minimal amount of data in the target task. In this case, controlling the learning trajectory is particularly important because the model is being trained on an unreliable sample of the true distribution of the target task, and we have to rely on early-stopping to prevent overfitting. We show that using the co-natural gradient during adaptation helps both at preserving source task performance and reach higher overall target task performance.

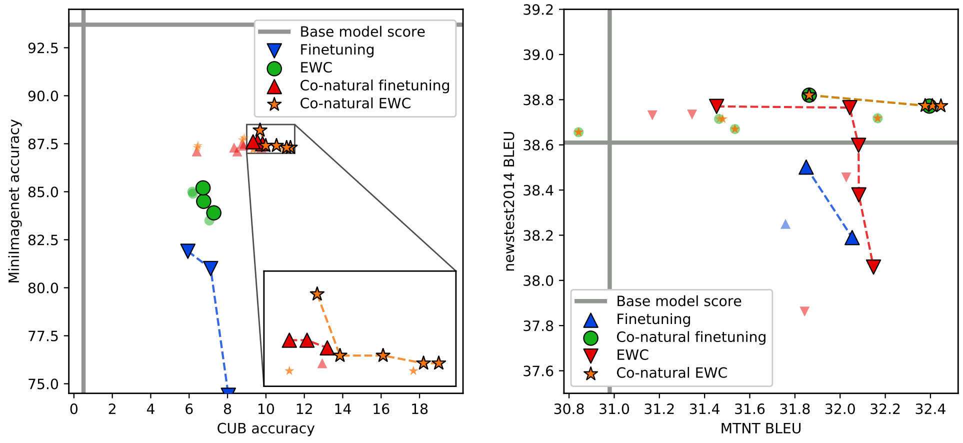
5.1 Experimental Setting

We perform experiments on two different scenarios:

Image classification We take MiniImagenet (a subset of the popular ImageNet (Deng et al., 2009) dataset[5]; Vinyals et al. (2016) as a source task and CUB (a 200-way birds species classification dataset; Welinder et al. (2010)) as a target task. To guarantee a strong base model despite the small size of MiniImageNet, we start off from a ResNet18 model (He et al., 2016) pretrained on the full ImageNet, which we retrofit to MiniImageNet by replacing the last fully connected layer with a separate linear layer regressed over the MiniImageNet training data. To simulate a low-resource setting, we sub-sample the CUB training set to 200 images ($\approx$ 1 per class). Scores for these tasks are reported in terms of accuracy.

Machine translation We consider adaptation of an English to French model trained on WMT15 (a dataset of parallel sentences crawled from parliamentary proceedings, news commentary and web page crawls; Bojar et al. (2015)) to MTNT (a dataset of Reddit comments; Michel & Neubig (2018)). Our model is a Transformer (Vaswani et al., 2017) pretrained on WMT15. Similarly to CUB, we

[5] Similarly to Omniglot, MiniImageNet was originally intended as a meta-learning benchmark and therefore its standard train/validation/test split consists of disjoint classes. We perform a custom transversal split so that the dataset can be used as a standard 100-way classification task. The accuracies reported here are not to be compared with the meta-learning literature.



(a) MiniImageNet to CUB adaptation

(b) WMT to MTNT fine-tuning

Figure 3: Low-resource adaptation results. The source (resp. target) task performance is represented on the vertical (resp. horizontal) axis. Pareto optimal configurations for each method are highlighted and the frontier is represented with dashed lines. The solid gray lines indicate the score of the original model trained on the source task.

simulate a low-resource setting by taking a sub-sample of 1000 sentence pairs as a training set. Scores for these two datasets are reported in terms of BLEU score.[6] (Papineni et al., 2002)

Here we do not allow any access to data in the source task when training on the target task. We compare four methods **Finetuning** (our baseline), **Co-natural finetuning**, **EWC** (which has been proven effective for domain adaptation, see Thompson et al. (2019)) and **Co-natural EWC**.

Given that different methods might lead to different trade-offs between source and target task performance, with some variation depending on the hyper-parameters (*e.g.* learning rate, regularization strength. . .), we take inspiration from Thompson et al. (2019) and graphically report results for all hyper-parameter configuration of each method on the 2 dimensional space defined by the score on source and target tasks[7]. Additionally, we highlight the Pareto frontier of each method *i.e.* the set of configurations that are not strictly worse than any other configuration for the same model.

5.2 Results

The adaptation results for both scenarios are reported in Figure 3. We find that in both cases, the co-natural gradient not only helps preserving the source task performance, but to some extent it also allows the model to reach better performance on the target task as well. We take this to corroborate our starting hypothesis: while introducing a regularizer does help, controlling the optimization dynamics actively helps counteract overfitting to the very small amount of training data, because the co-natural pre-conditioning makes it harder for stochastic gradient descent to push the model towards directions that would also hurt the source task.

6 Conclusion

We have presented the co-natural gradient, a technique that regularizes the optimization trajectory of models trained in a continual setting. We have shown that the co-natural gradient stands on its own as an efficient approach for overcoming catastrophic forgetting, and that it effectively complements and stabilizes other existing techniques at a minimal cost. We believe that the co-natural gradient — and more generally, trajectory regularization — can serve as a solid bedrock for building agents that learn without forgetting.

[6] We use sacrebleu (Post, 2018) with `-tok intl` as recommended by Michel & Neubig (2018).

[7] For CUB in particular we report the average accuracy of every configuration over 5 runs, each with a different 200-sized random subset of the data.

References

Hongjoon Ahn, Donggyu Lee, Sungmin Cha, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems (NIPS)*, 2019.

Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems (NIPS)*, 2019a.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems (NIPS)*, 2019b.

Shun-ichi Amari. Neural learning in structured parameter spaces-natural riemannian gradient. In *Proceedings of the 9th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 127–133, 1997.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the 10th Workshop on Statistical Machine Translation (WMT)*, pp. 1–46, 2015.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the 16th European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018a.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018b.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. *arXiv preprint arXiv:1902.10486*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Stephen T Grossberg. *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*, volume 70. Springer Science & Business Media, 1982.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350:1332–1338, 2015.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, 2016.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 6467–6476, 2017.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Paul Michel and Graham Neubig. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 543–553, 2018.

Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, 2002.

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

Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the 3rd Conference on Machine Translation (WMT)*, pp. 186–191, 2018.

Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2001–2010, 2017.

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

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 4535–4544, 2018.

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 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 (NAACL-HLT)*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 5998–6008, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 3630–3638, 2016.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 3987–3995, 2017.

A APPENDIX

A.1 DERIVATIONS

A.1.1 THE STANDARD GRADIENT UPDATE (EQUATION 3)

We derive the standard update in Eq. 3 by solving the Lagrangian $\mathbb{L}$ in Eq. 2 for δ. Given that its first and second derivatives are:

$$\begin{aligned}\nabla \mathbb{L} &= \nabla \mathcal{L}_T + 2\mu\delta \\ \nabla^2 \mathbb{L} &= 2\mu I\end{aligned}$$

the problem is trivially strictly convex and its global minimizer δ^* satisfies:

$$\nabla \mathbb{L}\big|_{\delta^*} = 0 \iff \delta^* = -\frac{1}{2\mu}\nabla \mathcal{L}_T$$

□

A.1.2 EQUIVALENCE OF THE HESSIAN OF THE KL DIVERGENCE AND THE FISHER INFORMATION MATRIX

To simplify notation, let us perform the change of variables $\theta + \delta \rightarrow x$. We show that the Hessian of the KL coincides with the Fisher on θ: in other words, $\nabla^2 \text{KL}(p_\theta \| p_x)\big|_{x=\theta} = \mathbb{E}_{p_\theta}[(\nabla \log p_\theta)(\nabla \log p_\theta)^\intercal]$. Under mild regularity assumptions[8], we can write:

$$\begin{aligned}\nabla^2 \text{KL}(p_\theta \| p_x) &= \underbrace{\nabla^2 \mathbb{E}_{p_\theta}[\log p_\theta]}_{=0} - \nabla^2 \mathbb{E}_{p_\theta}[\log p_x] \\ &= -\mathbb{E}_{p_\theta}[\nabla^2 \log p_x]\end{aligned}$$

Now note that $\nabla^2 \log p_x$ can be rewritten via standard derivatives manipulations as $\frac{\nabla^2 p_x}{p_x} - \frac{(\nabla p_x)(\nabla p_x)^\intercal}{p_x^2}$. This leads to:

$$\nabla^2 \text{KL}(p_\theta \| p_x) = -\mathbb{E}_{p_\theta}\left[\frac{\nabla^2 p_x}{p_x}\right] + \mathbb{E}_{p_\theta}\left[\left(\frac{\nabla p_x}{p_x}\right)\left(\frac{\nabla p_x}{p_x^2}\right)^\intercal\right]$$

When taken at θ, the first term evaluates to[9] :

$$\begin{aligned}\mathbb{E}_{p_\theta}\left[\frac{\nabla^2 p_x}{p_x}\right]\Bigg|_{x=\theta} &= \int p_\theta(z)\frac{\nabla^2 p_\theta(z)}{p_\theta(z)}dz \\ &= \int \nabla^2 p_\theta(z)dz \\ &= \nabla^2 \underbrace{\int p_\theta(z)dz}_{=1} = 0\end{aligned}$$

By using the identity $\frac{\nabla p_x}{p_x} = \nabla \log p_x$ and evaluating at $x = \theta$, the second term gives us:

$$\nabla^2 \text{KL}(p_\theta \| p_x)\big|_{x=\theta} = \mathbb{E}_{p_\theta}[(\nabla \log p_\theta)(\nabla \log p_\theta)^\intercal]$$

□

[8] Essentially allowing us to interchange derivatives and integrals.

[9] We abuse notation and write $\nabla p_x\big|_{x=\theta}$ as ∇p_θ

A.1.3 Obtaining the Co-natural Update (Equation 7)

We solve the Lagrangian from Eq. 6 in a similar fashion as in A.1.1. First we compute its gradient and Hessian:

$$\begin{aligned}\nabla \mathbb{L} &= \nabla \mathcal{L}_T + 2\mu\delta + 2\nu F_\theta^S \delta \\ &= \nabla \mathcal{L}_T + 2(\nu F_\theta^S + \mu I)\delta \\ \nabla^2 \mathbb{L} &= 2(\nu F_\theta^S + \mu I)\end{aligned}$$

While not as straightforwardly as the one in A.1.1, this problem is also strongly convex: indeed F_θ^S is positive semi-definite (as an expectation of PSD matrices) and the addition of μI ensures that $\nabla^2 \mathbb{L}$ is positive definite. We find the unique solution by solving:

$$\begin{aligned}\nabla \mathbb{L}\big|_{\delta^*} = 0 &\iff \nabla \mathcal{L}_T + 2(\nu F_\theta^S + \mu I)\delta^* = 0 \\ &\iff \delta^* = -[2\mu F_\theta^S + 2\nu I]^{-1} \nabla \mathcal{L}_T\end{aligned}$$

Set $\lambda := \frac{1}{2\mu}$ and $\alpha := \frac{\nu}{\mu}$ to get Eq. 7 □

A.2 Additional Experimental Settings for Continual Learning

This section is intended to facilitate the reproduction of our results. The full details can be found with our code at `anonymized_url`.

A.2.1 Split CIFAR

We split the dataset into 20 disjoint sub-tasks with each 5 classes, 2500 training examples and 500 test examples. This split, performed at random, is kept the same across all experiments, only the order of these tasks is changed. During continual training, we train the model for one epoch on each task with batch size 10, following the setup in Chaudhry et al. (2018b).

A.2.2 Omniglot

We consider each alphabet as a separate task, and split each task such that every character is present 12, 4 and 4 times in the training, validation and test set respectively (out of the 20 images for each character). During continual training, we train for 2500 steps with batch size 32 (in keeping with Schwarz et al. (2018)). We ignore the validation data and simply evaluate on the test set at the end of training.

A.2.3 Grid-search parameters

For each method, we perform grid-search over the following parameter values:

- Learning rate (all methods): 0.1, 0.03, 0.01
- Regularization strength (EWC, Co-natural EWC): 0.5, 1, 5
- Fisher damping coefficient (Co-natural finetuning, Co-natural EWC): 0,1.0,0.1 for Split CIFAR and 0,0.1,0.01 for Omniglot

For ER, we simply set the batch size to the same value as standard training (10 and 32 for Split CIFAR and Omniglot respectively).