
Unifying Importance Based Regularisation Methods for Continual Learning

Frederik Benzing

Department of Computer Science
ETH Zurich
8092 Zurich, Switzerland

Abstract

Continual Learning addresses the challenge of learning a number of different tasks sequentially. The goal of maintaining knowledge of earlier tasks without re-accessing them starkly conflicts with standard SGD training for artificial neural networks. An influential method to tackle this problem without storing old data are so-called regularisation approaches. They measure the importance of each parameter for solving a given task and subsequently protect important parameters from large changes. In the literature, three ways to measure parameter importance have been put forward and they have inspired a large body of follow-up work. Here, we present strong theoretical and empirical evidence that these three methods, Elastic Weight Consolidation (EWC), Synaptic Intelligence (SI) and Memory Aware Synapses (MAS), are surprisingly similar and are all linked to the same theoretical quantity. Concretely, we show that, despite stemming from very different motivations, both SI and MAS approximate the square root of the Fisher Information, with the Fisher being the theoretically justified basis of EWC. Moreover, we show that for SI the relation to the Fisher – and in fact its performance – is due to a previously unknown bias. On top of uncovering unknown similarities and unifying regularisation approaches, we also demonstrate that our insights enable practical performance improvements for large batch training.

1 INTRODUCTION

Despite considerable progress, many gaps between biological and machine intelligence remain. Animals, for example, flexibly learn new tasks throughout their lives, while at the same time maintaining robust memories of their previous knowledge. This ability conflicts with traditional training procedures for artificial neural networks, which overwrite previous skills when optimizing new tasks [McCloskey and Cohen, 1989, Goodfellow et al., 2013]. The field of continual learning is dedicated to mitigate this crucial shortcoming of machine learning. It exposes neural networks to a sequence of distinct tasks. When training for new tasks, the algorithm is not allowed to revisit old data. Nevertheless, it should retain previous skills and at the same time remaining flexible enough to incorporate new knowledge into the network.

An influential line of work to approach this challenge are so-called regularisation-based algorithms. They remain to be the main approach to continual learning without expanding the network or storing old data, which is a key challenge and a main motivation of the field [Farquhar and Gal, 2018]. The first regularisation-based method was introduced by [Kirkpatrick et al., 2017], who proposed the Elastic Weight Consolidation (EWC) algorithm. After training a given task, EWC measures the importance of each parameter for this task and introduces an auxiliary loss penalising large changes in important parameters. Naturally, this raises the question of how to measure ‘importance’. While EWC uses the diagonal Fisher Information, two main alternatives have been proposed: Synaptic Intelligence (SI, [Zenke et al., 2017]) aims to attribute the decrease in loss during training to individual parameters and Memory Aware Synapses (MAS, [Aljundi et al., 2018]) introduces a heuristic measure of output sensitivity. Together, these three approaches have inspired many further regularisation-based approaches, including combinations of them [Chaudhry et al., 2018],

refinements [Huszár, 2018, Ritter et al., 2018, Chaudhry et al., 2018, Yin et al., 2020], extensions [Schwarz et al., 2018, Liu et al., 2018, Park et al., 2019, Lee et al., 2020] and applications in different continual learning settings [Aljundi et al., 2019] as well as different domains of machine learning [Lan et al., 2019]. Almost every new continual learning method compares to at least one of the algorithms EWC, SI and MAS.

Despite their popularity and influence, basic practical and theoretical questions regarding these algorithms had previously been unanswered. Notably, it was unknown how similar these importance measures are. Additionally, for SI and MAS as well as their follow-ups there was no solid theoretical understanding of their effectiveness. Here, we close both these gaps through a theoretical analysis confirmed by a series of carefully designed experiments on standard continual learning benchmarks. Our main findings can be summarised as follows:

- (a) We show that SI’s importance approximation is biased and that the bias rather than SI’s original motivation is responsible for its performance. Further, also due to the bias, SI is approximately equal to the square root of the Fisher Information.
- (b) We show that MAS, like SI, approximately equals the square root of the Fisher Information.
- (c) Together, (a) and (b) unify the three main regularisation approaches and their follow ups by explicitly linking all of them to the same theoretically justified quantity – the Fisher Information. For SI and MAS-based algorithms this has the additional benefit of giving a more plausible theoretical explanation for their effectiveness.
- (d) Based on our precise understanding of SI, we propose an improved algorithm, Second-Order Synapses (SOS). We demonstrate that SOS outperforms SI in various regimes.

2 REVIEW OF REGULARISATION METHODS AND RELATED WORK

Here, we review importance based regularisation approaches (but see also [Li and Hoiem, 2017]). Additional related work is discussed in Appendix I.

Formal Description of Continual Learning. In continual learning we are given K datasets $\mathcal{D}_1, \dots, \mathcal{D}_K$ sequentially. When training a neural net with N parameters $\theta \in \mathbb{R}^N$ on dataset \mathcal{D}_k , we have no access to the previously seen datasets $\mathcal{D}_{1:k-1}$. However, at test time

the algorithms is tested on all K tasks and the average accuracy is taken as measure of the performance.

Common Framework for Regularisation Methods. Regularisation based approaches introduced in [Kirkpatrick et al., 2017, Huszár, 2018] protect previous memories by modifying the loss function L_k related to dataset \mathcal{D}_k . Let us denote the parameters obtained after finishing training on task k by $\theta^{(k)}$ and let $\omega^{(k)} \in \mathbb{R}^N$ be the parameters’ importances. When training on task k , regularisation methods use the loss

$$\tilde{L}_k = L_k + c \cdot \sum_{i=1}^N \omega_i^{(k-1)} \left(\theta_i - \theta_i^{(k-1)} \right)^2$$

where $c > 0$ is a hyperparameter. The first term L_k is the standard (e.g. cross entropy) loss of task k . The second term ensures that the parameters do not move away too far from their previous values. Usually, $\omega_i^{(k)} = \omega_i^{(k-1)} + \omega_i$, where ω_i is the importance for the most recently learned task k .

Elastic Weight Consolidation

[Kirkpatrick et al., 2017] uses the diagonal of the Fisher Information as importance measure. It is evaluated after training for a given task. To define the *Fisher Information Matrix* F , we consider a dataset \mathcal{X} . For each datapoint $X \in \mathcal{X}$ the network predicts a probability distribution \mathbf{q}_X over the set of labels L , where we suppress dependence of \mathbf{q}_X on θ for simplicity of notation. We denote the predicted probability of class y by $\mathbf{q}_X(y)$ for each $y \in L$. Let us assume that we minimize the negative log-likelihood $-\log \mathbf{q}_X(y)$ and write $g(X, y) = -\frac{\partial \log \mathbf{q}_X(y)}{\partial \theta}$ for its gradient.¹ Then the Fisher Information F is given by

$$F = \mathbb{E}_{X \sim \mathcal{X}} \mathbb{E}_{y \sim \mathbf{q}_X} [g(X, y)g(X, y)^T] \quad (1)$$

$$= \mathbb{E}_{X \sim \mathcal{X}} \sum_{y \in L} \mathbf{q}_X(y) \cdot g(X, y)g(X, y)^T. \quad (2)$$

Taking only the **diagonal Fisher** means $\omega_i(\text{EWC}) = \mathbb{E}_{X \sim \mathcal{X}} \mathbb{E}_{y \sim \mathbf{q}_X} [g_i(X, y)^2]$. Under the assumption that the learned label distribution \mathbf{q}_X is the real label distribution the Fisher equals the Hessian of the loss [Martens, 2014, Pascanu and Bengio, 2013].² Its use for continual learning has a clear Bayesian, theoretical interpretation [Kirkpatrick et al., 2017, Huszár, 2018].

Variational Continual Learning

[Nguyen et al., 2017] shares its Bayesian motivation

¹Gradients are assumed to be column vectors here and throughout.

²When we evaluate the importance of parameters for a given task, we only consider the loss of that task and gradients with respect to this loss. In our experiments, this will always be the cross-entropy. In particular, it does not include the regularisation loss.

with EWC, but uses principled variational inference in Bayesian neural networks. Its similarity to EWC and the fact that it uses ‘a smoothed version of the Fisher’ is already discussed in [Nguyen et al., 2017] and further formalised in [Loo et al., 2020]. We will therefore focus on the two algorithms below.

Memory Aware Synapses [Aljundi et al., 2018] heuristically argues that the sensitivity of the function output with respect to a parameter should be used as the parameter’s importance. This sensitivity is evaluated after training a given task. It can be measured with respect to either the logits or the probabilities. Here, we choose the latter option and describe the precise importance this leads to. In Appendix H we show that the resulting version of MAS has the same performance as the one described here, and, crucially, is also related to the Fisher Information.

Denoting, as before, the final layer of learned probabilities by \mathbf{q}_X , the MAS importance is

$$\omega_i(\text{MAS}) = \mathbb{E}_{X \sim \mathcal{X}} \left[\left| \frac{\partial \|\mathbf{q}_X\|^2}{\partial \theta_i} \right| \right],$$

MAS suggests using validation data to measure this importance. For a fair comparison, we measure importances on training data for all methods. We also confirmed that this choice does not affect our results.

Synaptic Intelligence [Zenke et al., 2017] approximates the contribution of each parameter to the decrease in loss and uses this contribution as importance. To formalise the ‘contribution of a parameter’, let us denote the parameters at time t by $\theta(t)$ and the loss by $L(t)$. If the parameters follow a smooth trajectory in parameter space, we can write the *decrease* in loss from time 0 to T as

$$L(0) - L(T) = - \int_{\theta(0)}^{\theta(T)} \frac{\partial L(t)}{\partial \theta} \theta'(t) dt \quad (3)$$

$$= - \sum_{i=1}^N \int_{\theta_i(0)}^{\theta_i(T)} \frac{\partial L(t)}{\partial \theta_i} \theta'_i(t) dt. \quad (4)$$

The i -th summand in (4) can be interpreted as the contribution of parameter θ_i to the decrease in loss. While we cannot evaluate the integral exactly, we can use a first-order approximation to obtain the importances. To do so, we write $\Delta_i(t) = (\theta_i(t+1) - \theta_i(t))$ for an approximation of $\theta'_i(t)dt$ and get

$$\tilde{\omega}_i(\text{SI}) = \sum_{t=0}^{T-1} \frac{\partial L(t)}{\partial \theta_i} \cdot \Delta_i(t). \quad (5)$$

Thus, we have $\sum_i \tilde{\omega}_i(\text{SI}) \approx L(0) - L(T)$. In addition, SI rescales its importances as follows³

$$\omega_i(\text{SI}) = \frac{\max\{0, \tilde{\omega}_i(\text{SI})\}}{(\theta_i(T) - \theta_i(0))^2 + \xi}. \quad (6)$$

³Note that the $\max(0, \cdot)$ is not part of the description

[Zenke et al., 2017] use additional assumptions to justify this importance. One of the them – using full batch gradient descent – is violated in practice and we will show that this has an important consequence.

Additional regularisation approaches. EWC, SI and MAS have inspired several follow ups. We presented a version of EWC due to [Huszár, 2018] and tested in [Chaudhry et al., 2018, Schwarz et al., 2018]. It is theoretically more sound and was shown to perform better. [Chaudhry et al., 2018] combine EWC with a ‘KL-rescaled’-SI. [Ritter et al., 2018] use a block-diagonal (rather than diagonal) approximation of the Fisher; [Yin et al., 2020] use the full Hessian matrix for small networks. [Liu et al., 2018] rotate the network to diagonalise the most recent Fisher Matrix, [Park et al., 2019] modify the loss of SI and [Lee et al., 2020] aim to account for batch-normalisation.

3 SYNAPTIC INTELLIGENCE, ITS BIAS AND THE FISHER

Here, we explain why SI is approximately equal to the square root of the Fisher, despite the apparent contrast between the latter and SI’s path integral. First, we identify the bias of SI when approximating the path integral. We then show that the bias, rather than the path integral, explains similarity to the Fisher as well as performance. We carefully validate each assumption of our analysis empirically.

3.1 Bias of Synaptic Intelligence

To calculate $\omega(\text{SI})$ (5), we need to calculate the product $p = \frac{\partial L(t)}{\partial \theta} \cdot \Delta(t)$ for each t . Evaluating the full gradient $\frac{\partial L}{\partial \theta}$ is too expensive, so SI uses a stochastic minibatch gradient. The estimate of p is biased since the same minibatch is used for the update Δ and the estimate of $\partial L / \partial \theta$.

We now give the calculations detailing this argument. For ease of exposition, let us assume vanilla SGD with learning rate 1 is used. Given a minibatch, denote its gradient estimate by $g + \sigma$, where $g = \partial L / \partial \theta$ denotes the full gradient and σ the noise. The update is $\Delta = g + \sigma$. Thus, p should be $p = g \cdot (g + \sigma)$. However, using $g + \sigma$, which was used for the parameter update, to estimate g results in $p_{\text{bias}} = (g + \sigma)^2$. Thus, the gradient noise introduces a bias of $\mathbb{E}[\sigma^2 + \sigma g] = \mathbb{E}[\sigma^2]$.

Unbiased SI (SIU). Having understood the bias, we can design an unbiased estimate by using two in-
in [Zenke et al., 2017]. However, we needed to include it to reproduce their results. A similar mechanism can be found in the official SI code.

dependent minibatches to calculate Δ and estimate g . We get $\Delta = g + \sigma$ and an estimate $g + \sigma'$ for g with independent noise σ' . We obtain $p_{no_bias} = (g + \sigma') \cdot (g + \sigma)$ which in expectation equals $p = g \cdot (g + \sigma)$. Based on this we define an unbiased importance measure

$$\tilde{\omega}_i(\text{SIU}) = \sum_{t=0}^{T-1} (g_t + \sigma'_t) \cdot \Delta(t).$$

Bias-Only version of SI (SIB). To isolate the bias, we can take the difference between biased and unbiased estimate. Concretely, this gives an importance which only measures the bias of SI

$$\tilde{\omega}_i(\text{SIB}) = \sum_{t=0}^{T-1} ((g + \sigma) - (g + \sigma'_t)) \cdot \Delta(t).$$

Observe that this estimate multiplies the parameter update $\Delta(t)$ with nothing but stochastic gradient noise. From the perspective of the SI path-integral, this should be meaningless and perform poorly. Our theory, detailed below, predicts differently.

3.2 Relation of SI's Bias to Fisher

The bias of SI depends on the optimizer used. The original SI-paper (and we) uses Adam [Kingma and Ba, 2014] and we now analyse the influence of this choice in detail; for other choices see Appendix B.

Recall that $\tilde{\omega}(\text{SI})$ is a sum over terms $\frac{\partial L(t)}{\partial \theta} \cdot \Delta(t)$, where $\Delta(t) = \theta(t+1) - \theta(t)$ is the parameter update at time t . Both terms, $\frac{\partial L(t)}{\partial \theta}$ as well as $\Delta(t)$, are computed using the same minibatch. Given a stochastic gradient $g_t + \sigma_t$, Adam keeps an exponential average of the gradient $m_t = (1 - \beta_1)(g_t + \sigma_t) + \beta_1 m_{t-1}$ as well as the squared gradient $v_t = (1 - \beta_2)(g_t + \sigma_t)^2 + \beta_2 v_{t-1}$. Ignoring minor normalisations, the parameter update is $\Delta(t) = \eta_t m_t / \sqrt{v_t}$, with learning rate $\eta_t = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Thus,

$$\begin{aligned} \frac{\partial L(t)}{\partial \theta} \Delta(t) &= \eta_t (1 - \beta_1) \frac{(g_t + \sigma_t)^2}{\sqrt{v_t}} + \eta_t \beta_1 \frac{(g_t + \sigma_t) m_{t-1}}{\sqrt{v_t}} \\ &\stackrel{\text{(A1)}}{\approx} \eta_t (1 - \beta_1) \frac{(g_t + \sigma_t)^2}{\sqrt{v_t}} \end{aligned} \quad (7)$$

Here, we made **Assumption (A1)** that the gradient noise is larger than the gradient, or more precisely: $(1 - \beta_1)\sigma_t^2 \gg \beta_1 m_{t-1} g_t$ (we ignore $\sigma_t m_{t-1}$ since $\mathbb{E}[\sigma_t m_{t-1}] = 0$ and since we average SI over many time steps). **(A1)** is equivalent to the bias of SI being larger than its unbiased part as detailed in Appendix E. Experiments in Section 3.3 and Appendix J provide strong support for this assumption. Next, we rewrite

$$\tilde{\omega}(\text{SI}) \stackrel{\text{(A1)}}{\approx} (1 - \beta_1) \sum_{t \leq T} \frac{\eta_t (g_t + \sigma_t)^2}{\sqrt{v_t}} \quad (8)$$

$$= \frac{1 - \beta_1}{\sqrt{v_T}} \sum_{t \leq T} \eta_t \sqrt{\frac{v_T}{v_t}} (g_t + \sigma_t)^2 \quad (9)$$

Now consider v_t . It is a decaying average of $(g_t + \sigma_t)^2$. When stochastic gradients become smaller during learning, v_t will decay so that $\sum \sqrt{v_T/v_t} \cdot (g_t + \sigma_t)^2$ will be a (unnormalised, not necessarily exponentially) decaying average of the squared gradient. Therefore, we make **Hypothesis (A2)**

$$\sum_{t \leq T} \sqrt{\frac{v_T}{v_t}} \cdot (g_t + \sigma_t)^2 \stackrel{\text{(A2)}}{\approx} v_T,$$

since both LHS & RHS are decaying averages of the squared gradient. We will validate **(A2)** in Section 3.3. Altogether, we obtain

$$\tilde{\omega}(\text{SI}) \approx \frac{1 - \beta_1}{\sqrt{v_T}} \sum_{t \leq T} \eta_t \sqrt{\frac{v_T}{v_t}} (g_t + \sigma_t)^2 \quad (10)$$

$$\stackrel{\text{(A2)}}{\approx} \frac{v_T}{\sqrt{v_T}} = \sqrt{v_T}.$$

To avoid confusion, we note that it may seem at first that a similar argument implies that SI is proportional to v_t rather than $\sqrt{v_t}$. We explain in Appendix D why this is not the case.

Relation of SI to Fisher (EWC). Recall that the exact form of the importance of SI depends on the optimizer used. Here, we used Adam following [Zenke et al., 2017]. The effect of SGD (+momentum) is very similar as discussed in Appendix B.

With Adam, the importance of SI is (due to its bias) roughly equal to $\sqrt{v_t}$. Note that v_t is a common approximation of the diagonal of the Hessian, e.g. [Graves, 2011, Khan et al., 2018, Aitchison, 2018], and recall that the Hessian is approximately equal to the Fisher (see Appendix C for a more detailed discussion). Thus, other than using different approximations of the Fisher, the only difference between EWC and SI is the square root of the latter.

Even though we are not aware of a theoretical justification of the square root in this setting, this result explicitly links the bias and importance measure of SI to second order information of the Hessian. On a related note, we show in Appendix G that in a different regime the square root of the Hessian can be a theoretically justified importance measure.

Influence of Regularisation. Note that gradients, second moment estimates and the update direction $\Delta(t)$

Table 1: **Test Accuracies on Permuted-MNIST and Split CIFAR** (Mean and std-err of average accuracy over 3 resp. 10 runs for MNIST resp. CIFAR). Experiments No. (1) & (3) explain SI’s and MAS’ performance. Exp (2) shows improvements of our SOS for large batch training and confirms our theoretical prediction.

No.	Algo.	P-MNIST	CIFAR	No.	Algo.	P-MNIST	CIFAR
(1)	SI	97.2±0.1	74.4±0.2	(3)	MAS	97.3±0.1	73.7±0.2
	SI Bias-Only	97.2±0.1	75.1±0.1		AF	97.4±0.1	73.4±0.1
	SI Unbiased	96.3±0.1	72.5±0.3		$\sqrt{\text{Fisher}}$	97.1±0.2	73.5±0.2
	SOS	97.3±0.1	74.1±0.2		Fisher (EWC)	97.1±0.2	73.1±0.2
(2)	SI(2048)	96.2±0.1	70.0±0.3				
	SOS(2048)	97.1 ± 0.1	74.4 ± 0.1				

of Adam will be influenced by the auxiliary regularisation loss. In contrast, the approximation of $\partial L/\partial\theta$ in (7) only relies on current task gradients (without regularisation). Thus, the relation between SI and $\sqrt{v_T}$ (which in this context is decaying average of task-only gradients), will be more noisy in the presence of large regularisation. This will be confirmed empirically in Section 3.3.

Practical Implications. A benefit of our derivation is that it makes the dependence of SI on the optimisation process explicit. Consider for example the influence of batch size. A priori one would not expect it to affect SI more than other methods. Given our derivation, however, one can see that a larger batch size reduces noise and thus bias, making (A1) less valid. Moreover, v_t will be a worse approximation of the Fisher (see Appendix C or e.g. [Khan et al., 2018]). Thus, if SI relies on its relation to second order information of the Fisher, then we expect it to suffer disproportionately from large batch sizes – a prediction we will confirm empirically below. Additionally, learning rate decay, choice of optimizer and trainingset size and difficulty could harm SI as discussed in Appendix B.

Improving SI: Second-Order Synapses (SOS).

Given our derivation, we propose to adapt SI to explicitly measure second order information contained in the Fisher. To this end, we propose the algorithm Second-Order Synapses (SOS). In its simplest form it uses $\omega(\text{SOS}) = \sqrt{v_t}$ as importance, where v_t is evaluated as described in the beginning of Section 3.2. Note that v_t can be measured independent of which optimizer is used. The value of v_t at the end of training a given task is taken as the algorithms importance. Importances from different tasks are summed, analogously to the other regularisation methods. Moreover, for larger batch sizes we introduce a similar, but provably better approximation of the Fisher detailed in Appendix C. We will confirm some benefits of SOS over SI empirically below and discuss additional advantages in

Table 2: **Test Accuracies on TinyImagenet.** Algorithms SI and SOS are evaluated on a version of ImageNet on different VGG architectures; results averaged across 3 seeds. Framework, hyperparameter selection and code directly taken from [De Lange et al., 2019] and only modified to support SOS. Results for EWC, MAS marked with an asterisk “*” are reported directly from [De Lange et al., 2019] without independent replication. Our replication of SI performs better than reported in [De Lange et al., 2019], despite using the original code.

SOS has substantial performance gains over SI.

Model	SI	SOS	EWC*	MAS*
Small	25.1 ± 4.6	44.3 ± 0.1	45.1	40.6
Base	46.0 ± 0.1	43.3 ± 0.3	42.4	46.9
Wide	40.0 ± 0.2	46.0 ± 0.1	31.1	45.1
Deep	21.6 ± 0.7	30.0 ± 0.1	29.1	33.6

Appendix B. Note also that SOS is computationally more efficient than MAS and EWC as it does not need a pass through the data after training a given task.

3.3 Empirical Investigation of SI, its bias and Fisher

Bias dominates SI. According to the motivation of SI the sum of importances over parameters $\sum_i \tilde{\omega}_i$ (SI) should track the decrease in loss $L(0) - L(T)$, see (4). Therefore, we investigated how well the summed importances of SI and its unbiased version SIU approximate the decrease in loss. We also include an approximation of the path integral which uses the full training set gradient $\partial L/\partial\theta$. The results in Figure 1 (left) show: (1) Using an unbiased gradient estimate and the full gradient gives almost identical sums, supporting the validity of the unbiased estimator.⁴ (2) The bias of

⁴Note that even the unbiased first order approximation of the path integral overestimates the decrease in loss. This is

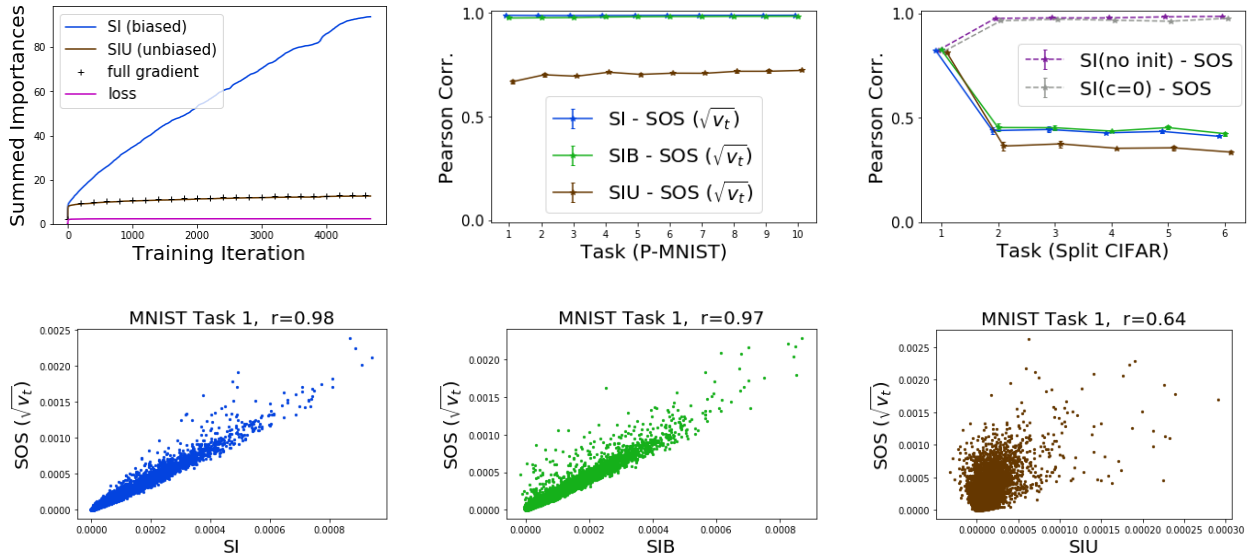


Figure 1: **SI, Bias and Square Root of Approximate Fisher v_t (SOS).**

Top Left: Summed Importances for SI and its unbiased version, showing that the bias dominates SI and that Assumption (A1) holds.

Top Center: Pearson correlations of SI, its bias (SIB), and unbiased version (SIU) with SOS, showing that relation between SI and SOS is strong and due to bias; confirming (A2). The same is shown by the scatter plots.

Top Right: Same as Mid but on CIFAR; additionally, relation between SOS and two SI-controls is shown: ‘no init’ does not re-initialise network weights after each task; ‘ $c = 0$ ’ has regularisation strength 0. This shows that strong regularisation weakens the tie between SI and SOS on Tasks 2-6 as explained by our theory.

Bottom: Scatter plots of SOS ($\sqrt{v_t}$) with SI (left), its bias-only SIB (middle) and its unbiased version SIU (right); showing 10^5 randomly sampled weights. A straight line through the origin corresponds to two importance measures being multiples of each other as was suggested by our derivation for SI and SOS, but not for SIU and SOS. Note that SIU has negative importances before rescaling in (6).

SI is 5-6 times larger than its unbiased so that SIU yields a considerably better approximation of the path integral.

Checking (A1). We point out that the bias, i.e. the difference between SI and SIU, in Figure 1 (left) is due to the term $(1 - \beta_1)\sigma_t^2$. Thus, the fact that the bias is considerably larger than the unbiased part is direct, strong evidence of Assumption (A1) (see Appendix E for full calculation). Results on CIFAR are analogous (Figure L.3).

Checking (A2): SI is almost equal to the Square Root of Approximate Fisher v_t . To check (A2), we compared $\omega(\text{SI})$ and $\omega(\text{SOS}) = \sqrt{v_t}$, see Figure 1 (mid, right). Correlations are almost equal to 1 on MNIST. The same holds on CIFAR Task 1, where the slight drop in correlation from around 0.99 to 0.9 is only due to the division in equation (6) (c.f. Appendix L). In summary, this shows that $\sqrt{v_t}$ is a very good approximation of SI. Correlations of SI to SOS on

consistent with findings that the loss has positive curvature [Jastrzebski et al., 2018, Lan et al., 2019].

CIFAR Task 2-6 decrease due to regularisation, see below.

Effect of Regularisation. The drop of correlations on CIFAR tasks 2-6 is due to large regularisation as explained theoretically in Section 3.2 and confirmed by two controls of SI with less strong regularisation: The first control simply sets regularisation strength to $c = 0$. The second control refrains from re-initialising the network weights after each task (exactly as in original SI, albeit with slightly worse validation performance than the version with re-initialisation). In the second setting the current parameters θ never move too far from their old value $\theta^{(k-1)}$, implying smaller gradients from the quadratic regularisation loss, and also meaning that a smaller value of $c = 0.5$ is optimal. We see that for both controls with weaker regularisation the correlation of SI to SOS is again close to one, supporting our theory.

Bias explains SI’s performance. We saw that SI’s bias is much larger than its unbiased part. But how does it influence SI’s performance? To quantify this,

we compared SI to its unbiased version SIU and the bias-only version SIB. Note that SIB is completely independent of the path integral motivating SI, only measures gradient noise and therefore should perform poorly according to SI’s original motivation. However, the results in Table 1(1) reveal the opposite: Removing the bias reduces performance of SI (SIU is worse), whereas isolating the bias does not affect or slightly improve performance. This demonstrates that SI relies on its bias, and not on the path integral, for its continual learning performance.

Bias explains Second-Order Information. We have seen that the bias, not the path integral, is responsible for SI’s performance. Our theory offers an explanation for this surprising finding, namely that the bias is responsible for SI’s relation to the Fisher. But does this explanation hold up in practice? To check this, we compare SI, its unbiased variant SIU and its bias-only variant SIB to the second order information $\omega(\text{SOS}) = \sqrt{v_T}$. The results in Figure 1 (top middle&right; bottom) show that SI, SIB are more similar to $\sqrt{v_T}$ than SIU, directly supporting our theoretical explanation.

Together the two previous paragraphs are very strong evidence that SI relies on the second order information contained in its bias rather than on the path integral.

SOS improves SI. We predicted that SI would suffer disproportionately from training with large batch sizes. To test this prediction, we ran SI and SOS with larger batch sizes of 2048, Table 1 (2). We found that, indeed, SI’s performance degrades by roughly 1% on MNIST and more drastically by 4% on CIFAR. In contrast, SOS’ performance remains stable.⁵ This validates our prediction and demonstrates that our improved understanding of SI leads to notable performance gains.

In addition, we carried out experiments on a version of ImageNet, following the protocol of a recent large-scale comparison of continual learning algorithms [De Lange et al., 2019]. The results are shown in Table 2. There is one setting (Base), where SI is slightly better than SOS and we are unsure why. Overall SOS clearly outperforms SI. This further supports the view that explicitly relying on the second-order information of the Fisher provides a more reliable, better performing algorithm.

⁵We emphasise that the difference of SI and SOS is *not* explained by a change in ‘training regime’ [Mirzadeh et al., 2020], since both algorithms use equally large minibatches.

4 MEMORY AWARE SYNAPSES (MAS) AND FISHER

Here, we explain why – despite its different motivation and like SI – MAS is approximately equal to the square root of the Fisher Information. We do so by first relating MAS to $\text{AF} := \mathbb{E}[|g + \sigma|]$ and then theoretically showing how this is related to the Fisher $F = \mathbb{E}[|g + \sigma|^2]$ under additional assumptions. As before, we check our theoretical derivations empirically.

4.1 Theoretical Relation of MAS and Fisher

We take a closer look at the definition of the importance of MAS. Recall that we use the predicted probability distribution of the network to measure sensitivity rather than logits. With linearity of derivatives, the chain rule and writing $y_0 = \text{argmax}_{\mathbf{q}} \mathbf{q}_X$, we see (omitting expectation over $X \sim \mathcal{X}$)

$$\begin{aligned} \left| \frac{\partial \|\mathbf{q}\|^2}{\partial \theta} \right| &= 2 \left| \sum_{y \in Y} \mathbf{q}(y) \frac{\partial \mathbf{q}(y)}{\partial \theta} \right| \stackrel{\text{(B1)}}{\approx} 2 \left| \mathbf{q}(y_0) \frac{\partial \mathbf{q}(y_0)}{\partial \theta} \right| \\ &= 2\mathbf{q}^2(y_0) \left| \frac{\partial \log \mathbf{q}(y_0)}{\partial \theta} \right| \end{aligned} \quad (11)$$

Here, we made **Assumption (B1)** that the sum is dominated by its maximum-likelihood label y_0 , which should be the case if the network classifies its images confidently, i.e. $\mathbf{q}(y_0) \gg \mathbf{q}(y)$ for $y \neq y_0$. Recall that the importance is measured at the end of training, so that this assumption is justified if the task has been learned successfully. Using the same assumption (B1) we get

$$\mathbb{E}_{y \sim \mathbf{q}_X} [|g + \sigma|] \stackrel{\text{(B1)}}{\approx} \mathbf{q}(y_0) |g + \sigma| = \mathbf{q}(y_0) \left| \frac{\partial \log \mathbf{q}(y_0)}{\partial \theta} \right|.$$

showing that the only difference between $\omega(\text{MAS})$ and $\mathbb{E}[|g + \sigma|]$ is a factor of $2\mathbf{q}_X(y_0)$. It is reasonable to make **Assumption (B2)** that this factor is approximately constant, since the model learned to classify training images confidently. This leads to **(B2)**: $\omega(\text{MAS}) \propto \text{AF} = \mathbb{E}[|g|]$. Note that even with a pessimistic guess that $\mathbf{q}_X(y_0)$ is in a range of 0.5 (rather inconfident) and 1.0 (absolutely confident), the two measures would be highly correlated.

Now, it remains to explore how $\mathbb{E}[|g + \sigma|]$ is related to the Fisher $F = \mathbb{E}[|g + \sigma|^2]$. The precise relationship between the two will depend on the distribution of $g + \sigma$. If, for example, gradients are distributed normally ($g + \sigma \sim \mathcal{N}(\mu, \Sigma)$ with $\Sigma_{i,i} \gg \mu_i$ (corresponding to the observation that the noise is much bigger than the gradients, which we have seen to be true above), then we obtain $F_i \approx \Sigma_{i,i}$ and, since $|g + \sigma|$ follows a folded normal distribution, we also have $\mathbb{E}[|g + \sigma|] \approx$

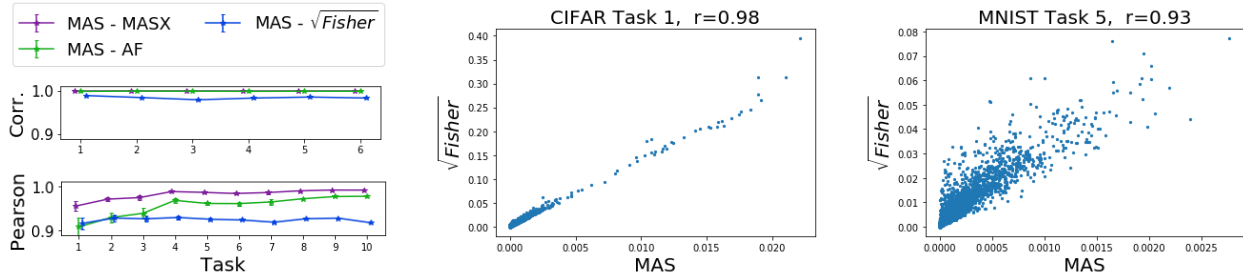


Figure 2: **Empirical Relation between MAS and Square Root of Fisher.**

Left: Summary of Pearson Correlations (top: CIFAR, bottom: MNIST), supporting Assumptions **(B1)**-**(B3)**. **Mid & Right:** Scatter plots of importance measures. Each point in the scatter plot corresponds to one weight of the net, showing 10^5 randomly sampled weights. A straight line through the origin corresponds to two importance measures being multiples of each other as was suggested by our theoretical analysis.

$c\sqrt{\Sigma_{ii}}$ with $c = \sqrt{2/\pi}$. Thus, in the case of a normal distribution with large noise, we have $\text{AF} = \mathbb{E}[|g|] \propto \sqrt{\text{F}}$. Note that the same conclusion arises in under different assumptions, too, e.g. assuming that gradients follow a Laplace distribution with $scale \gg mean$.

Relation of Fisher and MAS. The above analysis leads to **Hypothesis (B3)**: $\omega(\text{MAS}) \propto \sqrt{\text{Fisher}}$.

4.2 Empirical Relation of MAS and Fisher

First, to assess **(B1)** in (11) we compare LHS (MAS) and RHS (referred to as MASX) at the end of each task. We consistently find that the pearson correlations are almost equal to 1 strongly supporting **(B1)**, see Figure 2.

The correlations between $\omega(\text{MAS})$ and $\text{AF} = \mathbb{E}[|g + \sigma|]$ are similarly high, confirming **(B2)**.

The correlations between $\omega(\text{MAS})$ and the square root of the Fisher $\sqrt{\text{F}}$ are around 0.9 for Permuted MNIST and approximately 1 for all CIFAR task. This confirms our theoretical hypothesis **(B3)** that MAS is approximately equal to the square root of the Fisher.

In addition, to check whether similarity to the square root of the Fisher can serve as an explanation for performance of MAS, we ran continual learning algorithms based on MAS, AF and $\sqrt{\text{F}}$. The fact that these algorithms perform similarly, Table 1 (3), is in line with the claim that similarity to $\sqrt{\text{F}}$ provides a theoretically plausible explanation for MAS’ effectiveness.

5 EXPERIMENTAL SETUP

We outline the experimental setup, see Appendix F for details. It closely follows SI’s setting [Zenke et al., 2017]: In **domain-incremental Permuted MNIST** [Goodfellow et al., 2013] each of 10

tasks consists of a random (but fixed) pixel-permutation of MNIST and a **fully connected ReLU network** is used. The **task-incremental Split CIFAR 10/100** [Zenke et al., 2017] consists of six 10-way classification tasks, the first is CIFAR 10, and the other ones are extracted from CIFAR 100. The **keras default CIFAR 10 convolutional architecture** (with dropout and max-pooling) is used [keras,].

The only difference to the setup [Zenke et al., 2017] is that like [Swaroop et al., 2019], we usually re-initialise network weights after each task, observing better performance. Concretely, we run each method with and without re-initialisation (tuning the regularisation strength independently) and report the better results. On MNIST the improvements with this trick typically below .5% and on CIFAR vary between 0.% and around 2.%.

A known limitation of regularisation methods is that they are not applicable to class-incremental scenarios [van de Ven and Tolias, 2019, Hsu et al., 2018], and we do not include experiments in this setting. Code is available on github.⁶ For the TinyImageNet experiments in Table 2, we follow [De Lange et al., 2019], see also Appendix F.8.

6 DISCUSSION

We have investigated regularisation approaches for continual learning, which are the method of choice for continual learning without replaying old data or expanding the model. We have provided strong theoretical and experimental evidence that both MAS and SI approximate the square root of the Fisher Information. While the square root of the Fisher has no clear theoretical interpretation itself, our analysis makes explicit how

⁶https://github.com/freedbee/continual_regularisation

MAS and SI are related to second-order information contained in the Fisher. This provides a more plausible explanation of the effectiveness for MAS- and SI based algorithms. In addition, it shows how the three main regularisation methods are related to the same theoretically justified quantity, providing a unified view of these algorithms and their follow ups.

Moreover, our algorithm SIU to approximate SI's path integral can be used for the (non continual learning) algorithm LCA [Lan et al., 2019], which relies on an expensive approximation of the same integral. This opens up new opportunities to apply LCA to larger models and datasets.

For SI, our analysis included uncovering its bias. We found that the bias explains performance better than the path integral motivating SI. Our theory offers a sound, empirically confirmed explanation for this otherwise surprising finding. Beyond this theoretical contribution, by proposing the algorithm SOS we gave a concrete example how understanding the inner workings of SI leads to substantial performance improvements.

So far, studies providing a better understanding of regularisation algorithms and their similarities had been neglected. Our contribution fills this gap and, as exemplified above, proves to be a useful tool to predict, understand and improve these algorithms.

Acknowledgements

I thank Xun Zou, Florian Meier, Angelika Steger, Kalina Petrova and Asier Mujika for many helpful discussions and valuable feedback on this manuscript. Similarly, I am grateful for the constructive, detailed feedback of several anonymous reviewers. I am also thankful for AM's technical advice and the Felix Weisenberger Foundation's ongoing support.

References

- [Abadi et al., 2016] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.
- [Aitchison, 2018] Aitchison, L. (2018). Bayesian filtering unifies adaptive and non-adaptive neural network optimization methods. *arXiv preprint arXiv:1807.07540*.
- [Aljundi et al., 2018] Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. (2018). Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154.
- [Aljundi et al., 2019] Aljundi, R., Kelchtermans, K., and Tuytelaars, T. (2019). Task-free continual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11254–11263.
- [Chaudhry et al., 2018] Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. (2018). Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547.
- [Chaudhry et al., 2019] Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. (2019). Continual learning with tiny episodic memories. *arXiv preprint arXiv:1902.10486*.
- [De Lange et al., 2019] De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. (2019). Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*.
- [Farquhar and Gal, 2018] Farquhar, S. and Gal, Y. (2018). Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*.
- [Fernando et al., 2017] Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A. A., Pritzel, A., and Wierstra, D. (2017). Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*.
- [French, 1999] French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.
- [Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.
- [Golkar et al., 2019] Golkar, S., Kagan, M., and Cho, K. (2019). Continual learning via neural pruning. *arXiv preprint arXiv:1903.04476*.
- [Goodfellow et al., 2013] Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

- [Graves, 2011] Graves, A. (2011). Practical variational inference for neural networks. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 24, pages 2348–2356. Curran Associates, Inc.
- [Hsu et al., 2018] Hsu, Y.-C., Liu, Y.-C., Ramasamy, A., and Kira, Z. (2018). Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*.
- [Huszár, 2018] Huszár, F. (2018). Note on the quadratic penalties in elastic weight consolidation. *Proceedings of the National Academy of Sciences*, page 201717042.
- [Jastrzębski et al., 2017] Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. (2017). Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*.
- [Jastrzebski et al., 2018] Jastrzebski, S., Kenton, Z., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. (2018). On the relation between the sharpest directions of dnn loss and the sgd step length. *arXiv preprint arXiv:1807.05031*.
- [Kemker and Kanan, 2017] Kemker, R. and Kanan, C. (2017). Fearnnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*.
- [keras,] keras. Official keras cnn example. https://github.com/keras-team/keras/blob/master/examples/cifar10_cnn.py. Accessed: 2020-09-25.
- [Kessler et al., 2018] Kessler, S., Salas, A., Tan, V. W., Zohren, S., and Roberts, S. (2018). Practical bayesian learning of neural networks via adaptive optimisation methods. *arXiv preprint arXiv:1811.03679*.
- [Khan et al., 2018] Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018). Fast and scalable bayesian deep learning by weight-perturbation in adam. *arXiv preprint arXiv:1806.04854*.
- [Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [Kirkpatrick et al., 2017] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.
- [Kunstner et al., 2019] Kunstner, F., Balles, L., and Hennig, P. (2019). Limitations of the empirical fisher approximation. *arXiv preprint arXiv:1905.12558*.
- [Lan et al., 2019] Lan, J., Liu, R., Zhou, H., and Yosinski, J. (2019). Lca: Loss change allocation for neural network training. In *Advances in Neural Information Processing Systems*, pages 3614–3624.
- [Lee et al., 2020] Lee, J., Hong, H. G., Joo, D., and Kim, J. (2020). Continual learning with extended kronecker-factored approximate curvature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9001–9010.
- [Li et al., 2019] Li, X., Zhou, Y., Wu, T., Socher, R., and Xiong, C. (2019). Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. *arXiv preprint arXiv:1904.00310*.
- [Li and Hoiem, 2017] Li, Z. and Hoiem, D. (2017). Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.
- [Liu et al., 2018] Liu, X., Masana, M., Herranz, L., Van de Weijer, J., Lopez, A. M., and Bagdanov, A. D. (2018). Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2262–2268. IEEE.
- [Loo et al., 2020] Loo, N., Swaroop, S., and Turner, R. E. (2020). Generalized variational continual learning. *arXiv preprint arXiv:2011.12328*.
- [Lopez-Paz and Ranzato, 2017] Lopez-Paz, D. and Ranzato, M. (2017). Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476.
- [Martens, 2014] Martens, J. (2014). New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*.
- [McCloskey and Cohen, 1989] McCloskey, M. and Cohen, N. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24(C):109–165.
- [Mirzadeh et al., 2020] Mirzadeh, S. I., Farajtabar, M., Pascanu, R., and Ghasemzadeh, H. (2020). Understanding the role of training regimes in continual learning.
- [Nguyen et al., 2017] Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. (2017). Variational continual learning. *arXiv preprint arXiv:1710.10628*.

- [Parisi et al., 2019] Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*.
- [Park et al., 2019] Park, D., Hong, S., Han, B., and Lee, K. M. (2019). Continual learning by asymmetric loss approximation with single-side overestimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3335–3344.
- [Pascanu and Bengio, 2013] Pascanu, R. and Bengio, Y. (2013). Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*.
- [Ratcliff, 1990] Ratcliff, R. (1990). Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285.
- [Rebuffi et al., 2017] Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.
- [Ritter et al., 2018] Ritter, H., Botev, A., and Barber, D. (2018). Online structured laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems*, pages 3738–3748.
- [Schwarz et al., 2018] Schwarz, J., Luketina, J., Czarnecki, W. M., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. (2018). Progress & compress: A scalable framework for continual learning. *arXiv preprint arXiv:1805.06370*.
- [Shin et al., 2017] Shin, H., Lee, J. K., Kim, J., and Kim, J. (2017). Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999.
- [Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- [Srivastava et al., 2013] Srivastava, R. K., Masci, J., Kazerounian, S., Gomez, F., and Schmidhuber, J. (2013). Compete to compute. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 2310–2318. Curran Associates, Inc.
- [Swaroop et al., 2019] Swaroop, S., Nguyen, C. V., Bui, T. D., and Turner, R. E. (2019). Improving and understanding variational continual learning. *arXiv preprint arXiv:1905.02099*.
- [Thomas et al., 2020] Thomas, V., Pedregosa, F., Merriënboer, B., Manzagol, P.-A., Bengio, Y., and Le Roux, N. (2020). On the interplay between noise and curvature and its effect on optimization and generalization. In *International Conference on Artificial Intelligence and Statistics*, pages 3503–3513. PMLR.
- [van de Ven and Tolias, 2019] van de Ven, G. M. and Tolias, A. S. (2019). Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*.
- [von Oswald et al., 2019] von Oswald, J., Henning, C., Sacramento, J., and Grewe, B. F. (2019). Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*.
- [Yin et al., 2020] Yin, D., Farajtabar, M., and Li, A. (2020). Sola: Continual learning with second-order loss approximation.
- [Zenke et al., 2017] Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org.

Contents

1	INTRODUCTION	1
2	REVIEW OF REGULARISATION METHODS AND RELATED WORK	2
3	SYNAPTIC INTELLIGENCE, ITS BIAS AND THE FISHER	3
4	MEMORY AWARE SYNAPSES (MAS) AND FISHER	7
5	EXPERIMENTAL SETUP	8
6	DISCUSSION	8
A	Overview over Algorithms	13
B	Potential Advantages of SOS vs SI	14
C	Second Moment of Gradients, Hessian, Fisher and SOS	14
D	SI as Deacying Average	16
E	Exact Computation of Bias of SI with Adam	16
F	Experimental details	17
G	The Fisher and its square root for posterior variance	19
H	MAS based on logits	20
I	Related Work	21
J	Gradient Noise	22
K	Relative Performance of Regularisation Methods	22
L	More Experiments and Plots	23
A	Overview over Algorithms	

A tabluar overview of algorithms is given in Table A.1

B Potential Advantages of SOS vs SI

Here, we discuss some problems that SI may have and for which SOS offers a remedy due to its more principled nature. We also explain how the points discussed here could explain findings of a large scale empirical comparison of regularisation methods [De Lange et al., 2019] and leave testing these hypotheses to future work. We note that it seems difficult/impossible to even generate hypothesis for some of the findings in [De Lange et al., 2019] without our contribution. Further, if these hypotheses are true, the described shortcomings are addressed by SOS.

Table A.1: **Summary of Regularisation Methods and Related Baselines.**

Notation and Details: Algorithms on the top calculate importance ‘online’ along the parameter trajectory during training. Algorithms on the bottom calculate importance at the end of training a task by going through (part of) the training set again. Thus, the sum is over timesteps t (top) or datapoints X (bottom). N is the number of images over which is summed. For a datapoint X , \mathbf{q}_X denotes the predicted label distribution and $g(X, y)$ refers to the gradient of the negative log-likelihood of (X, y) . $\Delta(t) = \theta(t + 1) - \theta(t)$ refers to the parameter update at time t , which depends on both the current task’s loss and the auxiliary regularisation loss. Moreover, $(g_t + \sigma_t)$ refers to the stochastic gradient estimate of the current task’s loss (where g_t is the full gradient and σ_t the noise) given to the optimizer to update parameters. In contrast, $(g_t + \sigma'_t)$ refers to an independent stochastic gradient estimate.

Algorithms SI, SIU, SIB rescale their final importances as in equation (6) for fair comparison.

For description, justification and choice of α in SOS, see Appendix C.

Name	Parameter Importance $\omega(\cdot)$
SI	$\sum_t (g_t + \sigma_t) \Delta(t)$
SIU (SI-Unbiased)	$\sum_t (g_t + \sigma'_t) \Delta(t)$
SIB (SI Bias-only)	$\sum_t (\sigma_t - \sigma'_t) \Delta(t)$
SOS (simple)	$\frac{1 - \beta_2}{1 - \beta_2^{T+1}} \sum_{t \leq T} \beta_2^{T-t} (g_t + \sigma_t)^2$
SOS (2048, unbiased)	$\frac{1 - \beta_2}{1 - \beta_2^{T+1}} \sum_{t \leq T} \beta_2^{T-t} ((g_t + \sigma_t) - \alpha(g_t + \sigma'_t))^2$
Fisher (EWC)	$\frac{1}{N} \sum_X \mathbb{E}_{y \sim \mathbf{q}_X} [g(X, y)^2]$
AF	$\frac{1}{N} \sum_X \mathbb{E}_{y \sim \mathbf{q}_X} [g(X, y)]$
MAS	$\frac{1}{N} \sum_X \left \frac{\partial \ \mathbf{q}_X\ ^2}{\partial \theta} \right $

- Trainingset Size and Difficulty:** Recall that SI is similar to an unnormalised decaying average of $(g + \sigma)^2$. Since the average is unnormalised, the constant of proportionality between $\omega(\text{SI})$ and $\sqrt{v_t}$ will depend: (1) On the number of summands in the importance of SI, i.e. the number of training iterations and (2) On the speed of gradient decay, which may well depend on the training set difficulty. Thus, SI may underestimate the importance of small datasets (for which fewer updates are performed if the number of epochs is kept constant as e.g. in [Zenke et al., 2017, De Lange et al., 2019]) and also for easy datasets with fast gradient decay.
How would we expect this to affect performance? If easy datasets are presented first, then SI will undervalue their importance and thus forget them later on. If hard tasks are presented first, SI will overvalue their importance and make the network less plastic – as latter tasks are easy, this is not too big of a problem. So if easy tasks are presented first, we expect SI to perform worse on average and forget more than when hard tasks are presented first. Strikingly, both lower average accuracy as well as higher forgetting are exactly what [?, [Table 4] de2019continual observe for SI when easy tasks are presented first (but not for other methods).
- Learning Rate Decay:** In our derivation in Section 3.2 we assumed a constant learning rate $\eta_t = \eta_0$. If, however, learning rate decay is used, then this will counteract the importance of SI being a decaying average of $(g + \sigma)^2$; for a decaying learning rate more weight will be put on earlier gradients, which will likely make the estimate of the Fisher less similar to the Fisher at the end of training. This may explain why SI sometimes has worse performance than EWC, MAS in experiments of [De Lange et al., 2019]
- Choice of Optimizer:** Note that the bias and thus importance of SI depend on the optimizer. For example if we use SGD with learning rate η_t (with or without momentum), following the same calculations as before shows that $\omega(\text{SI}) \approx \sum_t \eta_t (g_t + \sigma_t)^2$. On the one hand, this is related to the Fisher so it may give good results, on the other hand it may overvalue gradients early in training, especially when combined with learning rate decay. See Appendix L for an experiment with SGD. [De Lange et al., 2019] use SGD with learning rate

decay, which again may be why SI sometimes performs worse than competitors.

In an extreme case, when we approximate natural gradient descent by preconditioning with the squared gradient, the SI importance will be constant across parameters, showcasing another, probably undesired, dependence of SI on the choice of optimizer.

4. **Effect of Regularisation:** We already discussed in the main part how strong regularisation influences SI's importance measure. In fact, on CIFAR tasks 2-6 many weights have negative importances, since the regularisation gradient points in the opposite direction of the task gradients and is larger, compare e.g. Figure 1 and L.1. Negative importances seem counterproductive in any theoretical framework.
5. **Batch Size:** We already predicted and confirmed that large batchsizes hurt SI and proposed a remedy to this issue, see also Appendix C.

C Second Moment of Gradients, Hessian, Fisher and SOS

As pointed out in the main paper, the second moment v_t of the gradient is a common approximation of the Hessian. Here, we briefly review why this is the case, why the approximation becomes worse for large batch sizes and show how to get a better estimate. We note that the relations between Fisher, Hessian, and squared gradients, which are recapitulated here, are discussed in several places in the literature, e.g. [Martens, 2014, Kunstner et al., 2019, Thomas et al., 2020].

One way to relate the squared gradients to the Hessian is through the Fisher⁷: The Fisher Information is an approximation of the Hessian, which becomes exact when the learned label distribution coincides with the real label distribution. The Fisher takes an expectation over the model's label distribution. A common approximation is to replace this expectation by the (deterministic) labels of the dataset. This is called the *Empirical Fisher* and it is a good approximation of the Fisher if the model classifies most (training) images correctly and confidently. To summarise, the Fisher is a good approximation of the Hessian under the assumption that the model's predicted distribution coincides with the real distribution and - under the same assumption - the empirical Fisher is a good approximation of the Hessian. In fact, taking a closer look at the derivations, it seems plausible that the empirical Fisher is a better approximation of the Hessian than the real Fisher, since - like the Hessian - it uses the real label distribution rather than the model's label distribution.

To avoid confusion and as pointed out by [Martens, 2014], we also note that the Empirical Fisher is not generally equal to the Generalised Gauss Newton matrix.

Note that the connection between squared gradients and the Hessian assumes that gradients are squared before averaging. Here, and in many other places e.g. [Khan et al., 2018, Aitchison, 2018], this is approximated by squaring after averaging over a mini-batch since this is easier to implement and requires less computation. We describe below why this approximation is valid for small batch sizes and introduce an improved and easy to compute estimate for large batch sizes. We have not seen this improved estimate elsewhere in the literature.

For this subsection, let us slightly change notation and denote the images by X_1, \dots, X_D and the gradients (with respect to their labels and the cross entropy loss) by $g + \sigma_1, \dots, g + \sigma_D$. Here, again g is the overall training set gradient and σ_i is the noise (i.e. $\sum_{i=1}^D \sigma_i = 0$) of individual images (rather than mini-batches). Then the Empirical Fisher is given by

$$\text{EF} = \frac{1}{D} \sum_{i=1}^D (g + \sigma_i)^2 = g^2 + \mathbb{E}[\sigma_k^2],$$

where k is uniformly drawn from $\{1, \dots, D\}$

Second Moment Estimate of Fisher. We want to compare EF to evaluating the squared gradient of a minibatch. Let i_1, \dots, i_b denote uniformly random, independent indices from $\{1, \dots, D\}$, so that X_{i_1}, \dots, X_{i_b} is a random minibatch of size b drawn with replacement. Let $g + \sigma$ be the gradient on this mini-batch. We then

⁷Alternatively, one can directly apply the derivation, which is used to relate Fisher and Hessian, to the squared gradient (Empirical Fisher).

have, taking expectations over the random indices,

$$\begin{aligned}
 \mathbb{E}[(g + \sigma)^2] &= \mathbb{E}\left[\frac{1}{b^2} \sum_{r,s=1}^b (g + \sigma_{i_r})(g + \sigma_{i_s})\right] \\
 &= \frac{b(b-1)}{b^2} \mathbb{E}[(g + \sigma_{i_1})(g + \sigma_{i_2})] + \frac{b}{b^2} \mathbb{E}[(g + \sigma_{i_1})^2] \\
 &= \frac{b-1}{b} g^2 + \frac{1}{b} \text{EF} \\
 &\approx \frac{1}{b} \text{EF}
 \end{aligned}$$

The last approximation is biased, but it is still a decent approximation as long as $\mathbb{E}[\sigma_k^2] \gg bg^2$. This explains why the second moment estimate of the Fisher gets worse for large batch sizes. Note that here σ_k refers to the noise of the gradient with batch size 1 (whereas σ is the noise of a mini-batch, which is b times smaller).

For an analysis assuming that the mini-batch is drawn without replacement, see e.g. [Khan et al., 2018]. Note that for a minibatch size of 2048 and a trainingset size of 60000, the difference between drawing with or without replacement is small, as on average there are only 35 duplicates in a batch with replacement, i.e. less than 2% of the batch.

C.1 Improved Estimate of Fisher for SOS.

We use the same notation as above, in particular σ_k refers to the gradient noise of a single randomly sampled image (not an entire minibatch). Let us denote by $g + \sigma$ and $g + \sigma'$ the gradient estimates obtained from two independent minibatches of size b (each sampled with replacement as above). Then for any $\alpha \in \mathbb{R}$, following the same calculations as above and slightly rearranging gives

$$\mathbb{E}\left[\left((g + \sigma) - \alpha(g + \sigma')\right)^2\right] = (1 - \alpha)^2 g^2 + \frac{1 + \alpha^2}{b} \mathbb{E}[\sigma_k^2],$$

where we used that each minibatch element is drawn independently and that $\mathbb{E}[\sigma_k] = 0$. Now, if $(1 - \alpha)^2 = \frac{1 + \alpha^2}{b}$, then the above expression is proportional to the Empirical Fisher $\text{EF} = g^2 + \mathbb{E}[\sigma_k^2]$. Since we rescale the regularisation loss with a hyperparameter, proportionality is all we need to get a strictly better (i.e. unbiased) approximation of the Empirical Fisher.

The above condition for α and b is satisfied when $\alpha = \frac{b + \sqrt{2b-1}}{b-1}$. In practice, we used $\alpha = 0$ for the experiments with batchsize 256 and $\alpha = 1$ for the experiments with batchsize 2048. Note that $\alpha = 1$ is always a good approximation when $\mathbb{E}[\sigma_k^2] \gg g^2$ (which we've seen to be the case). In our implementation SOS required calculating gradients on two minibatches, thus doubling the number of forward and backward passes during training. However, one could simply split the existing batch in two halves to keep the number of forward and backward passes constant. For large batch experiments we found introducing α necessary to obtain as good performance as with smaller batch sizes. This also supports our claim that SI's and SOS's importance measures rely on second order information contained in the Fisher.

D SI as Deacying Average

We concluded in the main paper that

$$\begin{aligned}
 \tilde{\omega}(\text{SI}) &\stackrel{\text{(A1)}}{\approx} (1 - \beta_1) \sum_{t \leq T} \frac{\eta_t (g_t + \sigma_t)^2}{\sqrt{v_t}} \\
 &= \frac{1 - \beta_1}{\sqrt{v_T}} \sum_{t \leq T} \eta_t \sqrt{\frac{v_T}{v_t}} (g_t + \sigma_t)^2 \\
 &\propto \frac{1}{\sqrt{v_T}} v_T
 \end{aligned} \tag{12}$$

arguing that $\sqrt{v_T/v_t}$ is increasing (i.e. v_t is decaying).

Upon first glance, it may seem that we could simply argue that $1/\sqrt{v_t}$ is increasing to conclude that

$$\tilde{\omega}(\text{SI}) \stackrel{\text{(A1)}}{\approx} (1 - \beta_1) \sum_{t \leq T} \eta_t \frac{1}{\sqrt{v_t}} (g_t + \sigma_t)^2 \propto v_T \quad (13)$$

However, note that in this case the constant of proportionality hidden in this notation depends on the gradient magnitude - and thus it will be different for parameters with different gradient magnitudes, so that the overall approximation is bad. To make this concrete, consider (13) and multiply the stochastic gradients by 2. This will increase $(g + \sigma_t)^2$ by a factor of 4, and $\sqrt{v_t}$ by a factor of 2, so the overall constant of proportionality will change by a factor of 2.

In contrast, in equation (12) the constant of proportionality will be unaffected, showing that this first is independent of gradient magnitude and thus more likely to be accurate.

E Exact Computation of Bias of SI with Adam

We claimed that the difference between SI and SIU (green and blue line) seen in Figure 1 (and also in Figures L.2, L.3) is due to the term $(1 - \beta_1)\sigma_t^2$. To see this, recall that for SI, we approximate $\frac{\partial L(t)}{\partial \theta}$ by $g_t + \sigma_t$, which is the same gradient estimate given to Adam. So we get

$$\text{SI:} \quad \frac{\partial L(t)}{\partial \theta} \Delta(t) = \frac{(1 - \beta_1)(g_t + \sigma_t)^2}{\sqrt{v_t} + \epsilon} + \frac{\beta_1(g_t + \sigma_t)m_{t-1}}{\sqrt{v_t} + \epsilon}.$$

For SIU, we use an independent mini-batch estimate $g_t + \sigma'_t$ for $\frac{\partial L(t)}{\partial \theta}$ and therefore obtain

$$\text{SIU:} \quad \frac{\partial L(t)}{\partial \theta} \Delta(t) = \frac{(1 - \beta_1)(g_t + \sigma'_t)(g_t + \sigma_t)}{\sqrt{v_t} + \epsilon} + \frac{\beta_1(g_t + \sigma'_t)m_{t-1}}{\sqrt{v_t} + \epsilon}.$$

Taking the difference between these two and ignoring all terms which have expectation zero (note that $\mathbb{E}[\sigma_t] = \mathbb{E}[\sigma'_t] = 0$ and that σ_t, σ'_t are independent of m_{t-1} and g_t) gives

$$\text{SI} - \text{SIU:} \quad (1 - \beta_1) \frac{\sigma_t^2}{\sqrt{v_t} + \epsilon}$$

as claimed.

Note also that in expectation SIU equals $(1 - \beta_1) \frac{g_t^2}{\sqrt{v_t} + \epsilon} + \beta_1 \frac{g_t m_{t-1}}{\sqrt{v_t} + \epsilon}$ so that a large difference between SI and SIU really means $(1 - \beta_1)\sigma_t^2 \gg (1 - \beta_1)g_t^2 + \beta_1 m_{t-1} g_t \approx \beta_1 m_{t-1} g_t$. The last approximation here is valid because $\beta_1 m_{t-1} \gg (1 - \beta_1)g_t$ which holds since (1) $\beta_1 \gg (1 - \beta_1)$ and (2) $\mathbb{E}[|m_{t-1}|] \geq g_t$ since $\mathbb{E}[|m_{t-1}|] \geq \mathbb{E}[m_{t-1}] \approx \mathbb{E}[|g_t|]$.

F Experimental details

F.1 Details of EWC and MAS related algorithms.

For EWC we calculate the ‘real’ Fisher Information as defined in the main article. For P-MNIST, we randomly sample 1000 training images (rather than iterating through the entire training set, which is prohibitively expensive). For Split CIFAR we use 500 random training images.

MAS, MAS-X and AF algorithms are also based on 1000 resp 500 random samples. When comparing these algorithms, we use the same set of samples for each algorithm.

F.2 Benchmarks

In **Permuted MNIST** [Goodfellow et al., 2013] each task consists of predicting the label of a random (but fixed) pixel-permutation of MNIST. We use a total of 10 tasks. As noted, we use a domain incremental setting, i.e. a single output head shared by all tasks.

In **Split CIFAR10/100** the network first has to classify CIFAR10 and then is successively given groups of 10 (consecutive) classes of CIFAR100. As in [Zenke et al., 2017], we use 6 tasks in total. Preliminary experiments on the maximum of 11 tasks showed very little difference. We use a task-incremental setting, i.e. each task has its output head and task-identity is known during training and testing.

F.3 Pre-processing

Following [Zenke et al., 2017], we normalise all pixel values to be in the interval $[0, 1]$ (i.e. we divide by 255) for MNIST and CIFAR datasets and use no data augmentation. We point out that pre-processing can affect performance, e.g. differences in pre-processing are part of the reason why results for SI on Permuted-MNIST in [Hsu et al., 2018] are considerably worse than reported in [Zenke et al., 2017] (the other two reasons being an unusually small learning rate of 10^{-4} for Adam in [Hsu et al., 2018] and a slightly smaller architecture).

F.4 Architectures and Initialization

For P-MNIST we use a fully connected net with ReLU activations and two hidden layers of 2000 units each. For Split CIFAR10/100 we use the default Keras CNN for CIFAR 10 with 4 convolutional layers and two fully connected layers, dropout [Srivastava et al., 2014] and maxpooling, see Table F.1. We use Glorot-uniform [Glorot and Bengio, 2010] initialization for both architectures.

F.5 Optimization

We use the Adam Optimizer [Kingma and Ba, 2014] with tensorflow [Abadi et al., 2016] default settings ($lr = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$) and batchsize 256 for both tasks like [Zenke et al., 2017]. On Permuted-MNIST we train each task for 20 epochs, on Split CIFAR we train each task for 60 epochs, also following [Zenke et al., 2017]. We reset the optimizer variables (momentum and such) after each task.

For the two experiments confirming our prediction regarding the effect of batchsize on SI, we made two changes on top of using a batch size of 2048: On P-MNIST we increased the learning rate by a factor of 8, since the ratio of batchsize and learning rate is thought to influence the flatness of minima (e.g. [Jastrzębski et al., 2017, Mirzadeh et al., 2020]). We did not tune or try other learning rates. On CIFAR, we tried the same learning rate adjustment but found that it led to divergence and we resorted back to the default 0.001. We observed that with a batch size of 2048, already on task 2, the network did not converge during 60 epochs (for tasks 2-6, one epoch corresponds to only 2 parameter updates with this batch size). We therefore increased the number of epochs to 600 on tasks 2-6, to match the number of parameter updates of Task 1 (CIFAR10).

F.6 SI & SOS details

Recall that we applied the operation $\max(0, \cdot)$ (i.e. a ReLU activation) to the importance measure of each individual task ((6) of main paper), before adding it to the overall importance. In the original SI implementation, this seems to be replaced by applying the same operation to the overall importances (after adding potentially negative values from a given task). No description of either of these operations is given in the SI paper. In light of our findings, our version seems more justified.

Somewhat naturally, the gradient $\frac{\partial L}{\partial \theta}$ usually refers to the cross-entropy loss of the current task and not the total loss including regularisation. For CIFAR we evaluated this gradient without dropout (but the parameter update was of course evaluated with dropout).⁸

For SOS, similarly to SI, we used the gradient evaluated without dropout on the cross-entropy loss of the current task for our importance measure $\sqrt{v_t}$.

When evaluated on benchmarks, SI, SIU, SIB all rescaled according to (6) to make the comparison as fair as possible. Scatter plots and correlations in the main text also use this rescaling, as this is the quantity used by the algorithms. In Appendix L we also include results before rescaling.

⁸We did not check the original SI code for what's done there and the paper does not describe which version is used.

Table F.1: CIFAR 10/100 architecture. Following [Zenke et al., 2017] we use the keras default architecture for CIFAR 10. Below, ‘Filt.’ refers to the number of filters of a convolutional layer, or respectively the number of neurons in a fully connected layer. ‘Drop.’ refers to the dropout rate. ‘Non-Lin.’ refers to the type of non-linearity used.

Table reproduced and slightly adapted from [Zenke et al., 2017].

Layer	Kernel	Stride	Filt.	Drop.	Non-Lin.
3x32x32 input					
Convolution	3x3	1x1	32		ReLU
Convolution	3x3	1x1	32		ReLU
MaxPool	2x2	2x2		0.25	
Convolution	3x3	1x1	64		ReLU
Convolution	3x3	1x1	64		ReLU
MaxPool	2x2	2x2		0.25	
FC			512	0.5	ReLU
Task 1: FC			10		softmax
...:FC			10		softmax
Task 6: FC			10		softmax

F.7 Hyperparameters

For all methods and benchmarks, we performed grid searches for the hyperparameter c and over the choice whether or not to re-initialise model variables after each task.

The grid of c included values $a \cdot 10^i$ where $a \in \{1, 2, 5\}$ and i was chosen in a suitable range (if a hyperparameter close to a boundary of our grid performed best, we extended the grid).

For CIFAR, we measured validation set performance based on at least three repetitions for good hyperparameters. We then picked the best HP and ran 10 repetitions on the test set. For MNIST, we measured HP-quality on the test set based on at least 3 runs for good HPs.

Additionally, SI and consequently SIU, SIB have rescaled importance (c.f. equation (6) from main paper). The damping term ξ in this rescaling was set to 0.1 for MNIST and to 0.001 for CIFAR following [Zenke et al., 2017] without further HP search.

All results shown are based on the same hyperparameters obtained – individually for each method – as described above. They can be found in Table F.2. We note that the difference between the HPs for MAS and MASX might seem to contradict our claims that the two measures are almost identical (they should require the same c in this case), but this is most likely due to similar performance for different HPs and random fluctuations. For example on CIFAR, MAS had almost identical validation performance for $c = 200$ (best for MAS) and $c = 1000$ (best for MASX) (74.2 ± 0.7 vs 74.1 ± 0.5).

Also for the other methods, we observed that usually there were two or more HP-configurations which performed very similarly. The precise ‘best’ values as found by the HP search and reported in Table F.2 are therefore subject to random fluctuations in the grid search.

F.8 Tiny ImageNet Experiments

For the experiments reported in Table 2, we followed exactly the setup form [De Lange et al., 2019] and did only minor modifications to the code to include SOS experiments. Note that also without these modifications, our results for SI (Table 2) do not match the ones reported in [De Lange et al., 2019], Table 3, (despite using the official code). We contacted the authors and they were uncertain where the discrepancy in results stems from.

F.9 Scatter Plot Data Collection

The correlation and scatter plots are based on a run of SI, where we in parallel to evaluating the SI importance measure also evaluated all other importance measures. Correlations in Figures 2 and 1 show data based on 5 repetitions (reporting mean and std-err as error bars) of this experiment and confirm that the correlations

Table F.2: Hyperparameter values for our experiments. See also maintext.

Algorithm	MNIST		CIFAR	
	c	re-init	c	re-init
SI	0.2	✓	5.0	✓
SI(2048)	1.0	✓	20.0	✓
SIU	2.0	✓	2.0	×
SIB	0.5	✓	5.0	✓
SOS	1.0	✓	500	✓
SOS(2048)	10	✓	2000	✓
MAS	500	×	200	✓
AF	200	×	1e3	✓
√Fisher	2	×	10	✓
EWC	1e3	×	5e4	✓
MAS2 (logits)	0.001	✓	0.1	✓

observed in the scatter plots are representative.

F.10 Computing Infrastructure

We ran experiments on one NVIDIA GeForce GTX 980 and on one NVIDIA GeForce GTX 1080 Ti GPU. We used tensorflow 1.4 for all experiments.

G The Fisher and its square root for posterior variance

Suppose that we know a locally optimal set of parameters θ_{opt} . Then we can use a Laplace approximation around θ_{opt} to find the likelihood of our parameters $\theta \sim \mathcal{N}(\theta_{opt}, K)$, where K is the (diagonal) inverse Hessian, or Fisher Information.

It is customary to use the final parameter point θ_{final} (after training) as approximation of θ_{opt} , which also leads to the EWC algorithm. However, if we have uncertainty estimates for θ_{opt} we can (and should?) incorporate these into the model. Assuming $\theta_{opt} \sim \mathcal{N}(\theta_{final}, R)$, with $p(\theta) = \int p(\theta | \theta_{opt}) \cdot p(\theta_{opt}) d\theta_{opt}$, we get

$$\theta \sim \mathcal{N}(\theta_{final}, K + R)$$

where we assumed that the inverse Hessian K is constant across values of θ_{opt} (which is an assumption already made in [Huszár, 2018] providing a Bayesian justification for EWC).

Recently, [Aitchison, 2018] presented a Kalman filter model for optimization, whose uncertainty estimates for θ_{opt} , recover the Adam Optimizer [Kingma and Ba, 2014] and we now show how to plug in this model in the above equations. Ideas from [Kessler et al., 2018] are also closely related.

For sake of simplicity we take $\sqrt{\eta^2/\langle g^2 \rangle}$ as posterior variance of θ_{opt} , where $\langle g^2 \rangle = v_T$ is the exponentially decaying average of the second moment of the stochastic gradients. In the framework of [Aitchison, 2018] this corresponds to a uniform prior (which, judging from the absence of weight decay, is used by EWC, SI).⁹

Now, using this framework, we plug in suitable values for K, R . Let’s denote the mini-batch size by b and the exponentially decaying average of the gradient’s second moment by $\langle g^2 \rangle$. Then the Hessian is approximately $b \cdot \langle g^2 \rangle$, giving $K = 1/(b \cdot \langle g^2 \rangle)$. As discussed above, the variance R of θ_{opt} is given by $\sqrt{\eta^2/\langle g^2 \rangle}$, where $\eta = 0.001$. It may seem that we should also multiply g^2 by b since g^2 serves as an approximation for the Hessian here, too. However, the batch size was already compensated for by the choice of η as long as we use a ‘standard’ batch size, let’s say $b \sim 100$ (see [Aitchison, 2018] and below).

Altogether, we get a posterior variance of the likelihood of θ of

$$K + R = \frac{1}{b\langle g^2 \rangle} + \frac{\eta}{\sqrt{\langle g^2 \rangle}}$$

⁹Note that in addition to a uniform prior, this also relies on the assumption that gradients are not too large, by which we mean that $\sqrt{v_T}$ is not much bigger than ~ 1000 , since otherwise the approximation in equation (47) of [Aitchison, 2018] becomes inaccurate due to large values of λ_{post} .

Using the inverse variance as an importance measure for continual learning, we get

$$\frac{1}{K+R} = \frac{b\langle g^2 \rangle}{1 + b\eta\sqrt{\langle g^2 \rangle}}.$$

For $b\eta\sqrt{\langle g^2 \rangle} > 1$ (or $\sqrt{\langle g^2 \rangle} \sim 10$ and $b \sim 100$), this importance (as well as the posterior variance) is approximated better by the square-root $\sqrt{\langle g^2 \rangle}$ than by $\langle g^2 \rangle$ itself. As already pointed out in the main text, this is not quite the regime we operate in on Permuted-MNIST and Split CIFAR with the given architectures.

G.1 Dependence on Batch Size

This subsection assumes familiarity with [Aitchison, 2018] and specifically, the argument used there to choose η .

To make the dependence on the minibatch size clearer, we can use $b\langle g^2 \rangle$ everywhere as approximation of the Hessian, leading to $\sqrt{\eta'^2/(b\langle g^2 \rangle)}$ as posterior variance of θ_{opt} . Assuming $b \sim 100$ we should then set $\eta' = 0.01$ instead. This results in

$$K+R = \frac{1}{b\langle g^2 \rangle} + \frac{\eta'}{\sqrt{b\langle g^2 \rangle}}$$

and

$$\frac{1}{K+R} = \frac{b\langle g^2 \rangle}{1 + \eta'\sqrt{b\langle g^2 \rangle}}.$$

H MAS based on logits

As pointed out in Section 2, there are two versions of MAS. Here, we describe results related to the version based on logits. We note that this version has the potentially undesirable feature of not being invariant to reparametrisations: If, for example, we add a constant c to all the logits, this does not change predictions (or the training process), but it does change the logit-based MAS importance.

We found identical performance as for the both versions of MAS (of course tuning HPs individually): 97.2 ± 0.1 on P-MNIST and 73.9 ± 0.2 on CIFAR for logit based MAS, as compared to 97.3 ± 0.1 and 73.7 ± 0.2 for the version based on the probability-distribution output. For both benchmarks, the difference based on 10 repetitions was not statistically significant ($p > 0.4$ for both benchmarks, t-test).

Next, we investigated the relation between logit based MAS and the square root of the Fisher. On CIFAR we found correlations very close to 1 for all pairs and on MNIST correlations were slightly weaker, see Figure H.1. Note that on MNIST the correlations between e.g. MAS and AF are similar to the correlation between AF and SI, Figure H.1 (right). We also show how AF depends on sample size, comparing 1000 (used by MAS) to 30000 samples on MNIST and 500 to 5000 on CIFAR, see Figure H.1 (right).

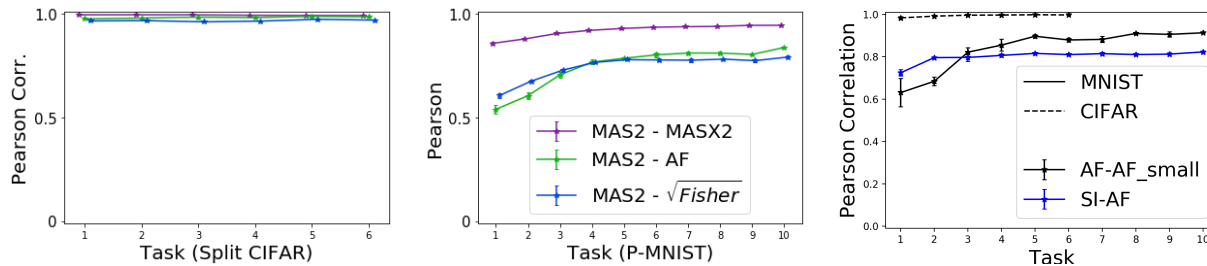


Figure H.1: **Empirical Relation between MAS (logits) and Square Root Fisher.**

Left & Mid: Same as Figure 2, left, but using MAS based on logits rather than output distribution.

Right: Comparing AF based on different sample sizes. 1000 vs 30000 on MNIST and 500 vs 5000 on CIFAR. Comparing SI, which relies on gradients of all 60000 images, to AF using 30000 images.

I Related Work

The problem of catastrophic forgetting in neural networks has been studied for many decades [McCloskey and Cohen, 1989, Ratcliff, 1990, French, 1999]. In the context of deep learning, it received more attention again [Goodfellow et al., 2013, Srivastava et al., 2013].

We now review the broad body of continual learning algorithms. Following [Parisi et al., 2019], they are often categorised into regularisation-, replay- and architectural approaches.

Regularisation methods have been reviewed in the main paper. We note that [Li and Hoiem, 2017] is often also called a regularisation method, while being conceptually different from the ones described previously.

Replay methods refer to algorithms which either store a small sample or generate data of old distributions and use this data while training on new methods [Rebuffi et al., 2017, Lopez-Paz and Ranzato, 2017, Shin et al., 2017, Kemker and Kanan, 2017]. These approaches can – but certainly do not have to – be seen as investigating how far standard i.i.d.-training can be relaxed towards the (highly non-i.i.d.) continual learning setting without losing too much performance. They are interesting, but usually circumvent the original motivation of continual learning to maintain knowledge *without* accessing old distributions. Intriguingly, the most effective way to use old data appears to be simply replaying it, i.e. mimicking training with i.i.d. batches sampled from all tasks simultaneously [Chaudhry et al., 2019].

Architectural methods extend the network as new tasks arrive [Fernando et al., 2017, Li et al., 2019, Schwarz et al., 2018, Golkar et al., 2019, von Oswald et al., 2019]. This can be seen as a study of how old parts of the network can be effectively used to solve new tasks and touches upon transfer learning. Typically, it avoids the challenge of integrating new knowledge into an existing networks. Finally, [van de Ven and Tolias, 2019, Hsu et al., 2018, Farquhar and Gal, 2018] point out that different continual learning scenarios and assumptions with varying difficulty were used across the literature.¹⁰

J Gradient Noise

Here, we quantitatively assess the noise magnitude outside the continual learning context. Recall that Figure 1 (left) from the main paper, as well as Figures L.2 and L.3 already show that the noise dominates the SI importance measure, which indicates that the noise is considerably larger than the gradient itself.

To obtain an assessment independent of the SI continual learning importance measure, we trained our network on MNIST as described before, i.e. a ReLU network with 2 hidden layers of 2000 units each, trained for 20 epochs with batch size 256 and default Adam settings. At each training iteration, on top of calculating the stochastic mini-batch gradient used for optimization, we also computed the full gradient on the entire training set and computed the noise – which refers to the squared ℓ_2 distance between the stochastic mini-batch gradient and the full gradient – as well as the ratio between noise and gradient, measured as the ratio of squared ℓ_2 norms. The results are shown in Figure J.1 (top). In addition, we computed the fraction of iterations in which the ratio between noise and squared gradient norm is above a certain threshold, see Figure J.1 (bottom).

K Relative Performance of Regularisation Methods

Several papers [van de Ven and Tolias, 2019, Hsu et al., 2018, Farquhar and Gal, 2018] played an important role in recognising that different continual learning algorithms had been evaluated in settings with varying difficulties. These papers also greatly clarified these settings, making an important contribution for future work.

Here, we point out that some of the results (and thus conclusions) of the experiments in [Hsu et al., 2018] (these were the only ones we investigated more closely) are possibly specific to settings and hyperparameters used there.

For example, it is found there that EWC, MAS, SI perform worse than an importance measure which assigns the same importance to all parameters (called ‘L2 regularisation’). This is in contrast to results reported in the EWC paper [Kirkpatrick et al., 2017], who found the L2 baseline to perform worse than EWC. We also attempted to reproduce the L2 results and tried two versions of this approach, one with constant importance for all tasks and one with importance increasing linearly with the number of tasks. The latter approach worked better, but

¹⁰We critically review and question some of their experimental results below.

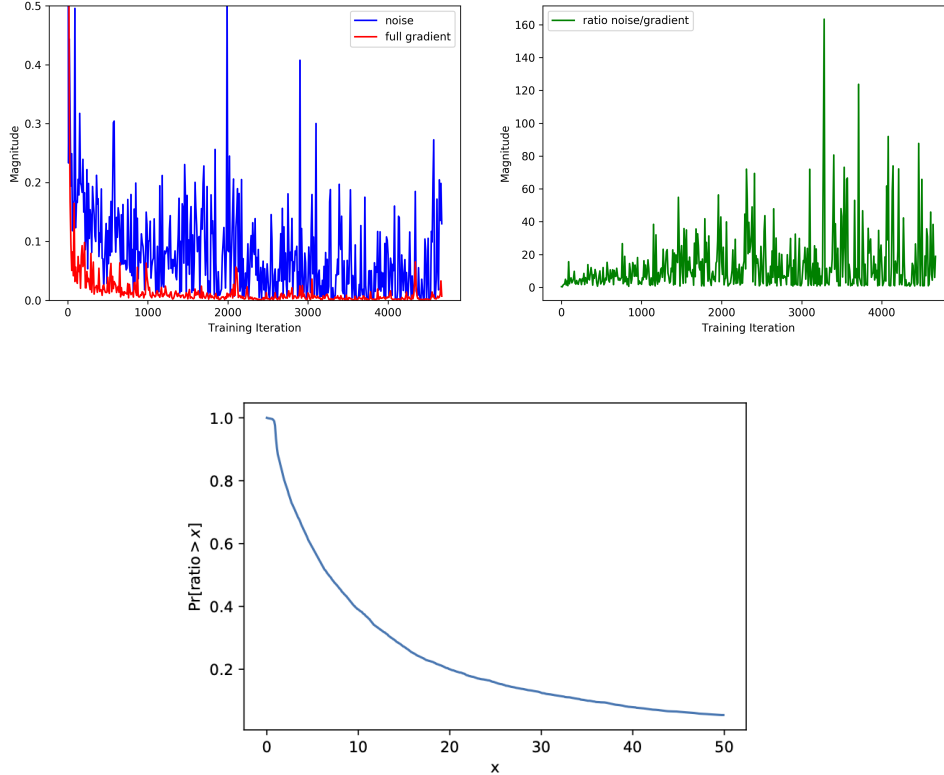


Figure J.1: **Top:** Gradient noise, measured as squared ℓ_2 distance between full training set gradient and the stochastic mini-batch gradient with a batch size of 256. ‘Full gradient’ magnitude is also measured as squared ℓ_2 norm.

Data obtained by training a ReLu network with 2 hidden layers of 2000 hidden units for 20 epochs with default Adam settings. Only every 20-th datapoint shown for better visualisation.

Bottom: Same data. y -value shows fraction of training iterations in which the ratio between mini-batch noise and full training set gradient was at least x -value. In particular the batch-size was 256. ‘Ratio’ refers to the ratio of squared ℓ_2 norms of the respective values.

still considerably underperformed EWC, MAS, SI on Permuted-MNIST (88% vs 97%). We didn’t run these experiments on CIFAR. An alternative importance measure based on the weights’ magnitudes after training was better, but still could not compete with SI, EWC, MAS (<95% vs >97%) showing that these importance measure are more use- and meaningful than naïve baselines.

Additionally, in other work we found that for domain incremental settings (again, these were the only ones we investigated), results of baselines EWC and SI, but also of fine-tune (a baseline which takes no measures to prevent catastrophic forgetting) can be improved, often by more than 10% compared to previously mentioned reports [Hsu et al., 2018, van de Ven and Tolias, 2019]. This also implies that the difference between well set-up regularisation approaches and for example repaly-methods is not nearly as big as previously thought. The differences between these results are explained by different factors (we explicitly tested each factor): hyperparameters (for example using Adam with the standard learning rate 0.001 improves performance over the learning rate 0.0001 used previously), initialisation (Glorot Uniform Initialisation works considerably better for SI than pytorch standard initialisation for linear layers, which reveals unexpected defaults upon close investigation), training duration (regularisation approaches usually notably benefit from longer training, this is reported in [Swaroop et al., 2019], and we also found that SI’s performance on P-MNIST can be improved to 98% by training for 200 rather than 20 epochs; [Hsu et al., 2018] use very short training times), data-preprocessing (depending on the setting different normalisations have different, non-negligible impacts).

L More Experiments and Plots

Here, we include additional plots and experiments.

L.1 SI with SGD

We tested SGD+momentum on Permuted MNIST for SI. We found that SI has similar performance as before and again is better SIU (which is also similar to before). However, training was more unstable and occasionally diverged for both SI and SIU, probably due to the ill-conditioned optimization caused by the regularisation loss, so that we did not run SGD experiments on CIFAR.

L.2 Bias of SI on All tasks

Here, we report the magnitude of the bias of SI on all tasks, analogously to Figure 1, left panel. The observation that most of the SI importance is due to its bias is consistent across datasets and tasks, see Figures L.2 and L.3. Intriguingly, on CIFAR we find that the unbiased approximation of SI slightly underestimates the decrease in loss in the last task, suggesting that strong regularisation pushes the parameters in places, where the cross-entropy of the current task has negative curvature.

We did not conduct the analogue of experiment corresponding to the controls in Figure 1, right panel, on MNIST since there the influence of regularisation was already weak.

L.3 Comparison of SI, SIU, SIB to SOS before rescaling

The comparisons between SI, SIU, SIB and SOS in the main paper were obtained before rescaling in (6).

Results before rescaling are shown and discussed in Figure L.1, also confirming that correlation between SOS and SI is due to the bias of SI.

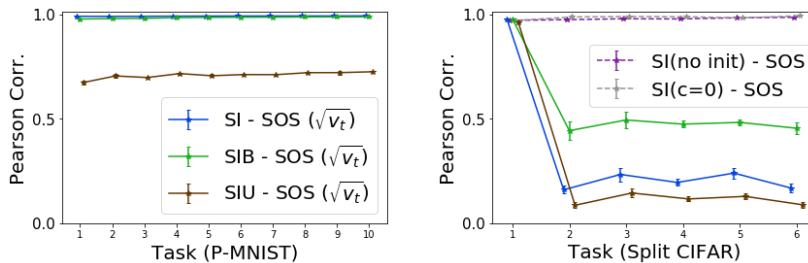


Figure L.1: Analogous to Figure 1 from main paper, comparing SI, SIU, SIB to SOS, but before applying the rescaling in (6) to SI, SIU, SIB (but not SOS). Note that rescaling on MNIST has close to no effect, which is due to almost all importances being non-negative and the denominator in (6) being almost equal to the damping term $\xi = 0.1$. On CIFAR, rescaling increases correlations on Tasks 2-5, mostly due to the $\max(\cdot, 0)$ operation. On Task 1, correlation to SOS is decreased by rescaling, as expected, due to division in (6). There are no negative importances after first task, due to absence of the regularisation term.

Note that it may seem surprising that on the first task of Split CIFAR (i.e. CIFAR10) all of SIU, SIB, SI have similarly high correlations to SOS in this figure as well as Figure 1. However, similarity between SIU before rescaling and SOS in this situation is explained by the weights with largest importance: If we remove the 5% of weights (results similar for 1%) which have highest SIU importance, correlation between SOS, SIU drops to 0.5, but for SI resp. SIB the same procedure yields 0.9. This indicates that for the first task of CIFAR there is a small fraction of weights with large importances, which are dominated by the gradient rather than the noise. The remainder of weights is in accordance with our previous observations and dominated by noise, recall also Figure L.3.

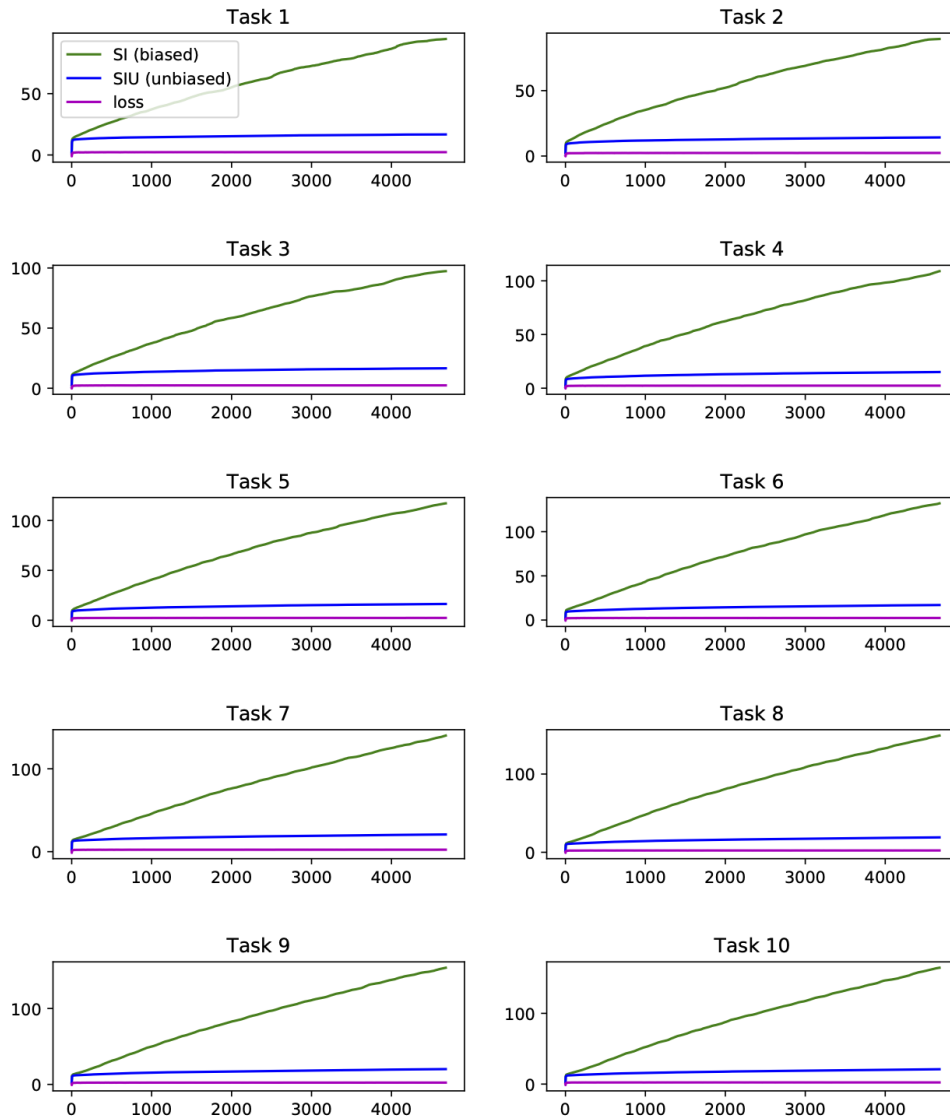


Figure L.2: Summed importances for all P-MNIST tasks for SI and its unbiased version. Analogous to Figure 1 (left) from main paper. Note and excuse changed color coding of SI, SIU.

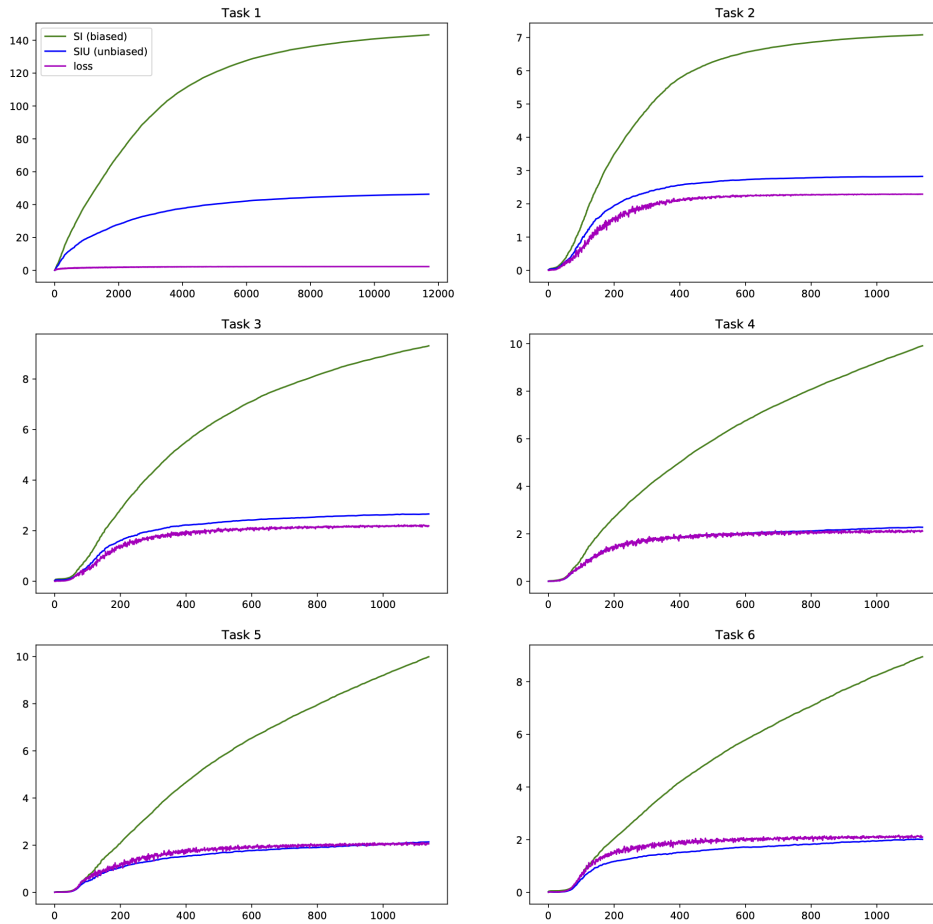


Figure L.3: Summed importances for all Split CIFAR 10/100 tasks for SI and its unbiased version. Analogous to Figure 1 (left) from main paper. Note and excuse changed color coding of SI, SIU.