Supplementary Material: Continual Learning Through Synaptic Intelligence

Friedemann Zenke [*1] Ben Poole [*1] Surya Ganguli [1]

A. Split CIFAR-10/100 CNN architecture

For our CIFAR-10/100 experiments, we used the default CIFAR-10 CNN from Keras:

Operation	Kernel	Stride	Filters	Dropout	Nonlin.
3x32x32 input					
Convolution	3×3	1×1	32		ReLU
Convolution	3×3	1×1	32		ReLU
MaxPool		2×2		0.25	
Convolution	3×3	1×1	64		ReLU
Convolution	3×3	1×1	64		ReLU
MaxPool		2×2		0.25	
Dense			512	0.5	ReLU
Task 1: Dense			m		
. . . : Dense			m		
Task μ: Dense			m		

Table 1. Split CIFAR10/100 model architecture and hyperparameters. m: number of splits.

B. Additional split CIFAR–10 experiments

As an additional experiment, we trained a CNN (4 convolutional, followed by 2 dense layers with dropout; cf. main text) on the split CIFAR-10 benchmark. We used the same multi-head setup as in the case of split MNIST using Adam ($\eta = 1 \times 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, minibatch size 256). First, we trained the network for 60 epochs on the first 5 categories (Task A). At this point the training accuracy was close to 1. Then the optimizer was reset and the network was trained for another 60 epochs on the remaining 5 categories (Task B). We ran identical experiments for both the control case ($c = 0$) and the case in which consolidation was active ($c > 0$). All experiments were repeated $n = 10$ times to quantify the uncertainty on the validation set accuracy.

After training on both Task A and B, the network with consolidation performed significantly better on both tasks than the control network without consolidation (Fig. 1). While the large performance difference on Task A can readily be explained by the fact that consolidation alleviates the problem of catastrophic forgetting — the initial motivation for our model — the small but significant difference ($\approx 4.5\%$) in validation accuracy on Task B suggests that consolidation also improves transfer learning. The network without consolidation is essentially fine-tuning a model which has been pre-trained on the first five CIFAR-10 categories. In contrast to that, by leveraging the knowledge about the optimization of Task A stored at the individual synapses, the network with consolidation solves a different optimization problem which makes the network generalize better on Task B. This significant effect was observed consistently for different values of c in the range $0.1 < c < 10$.

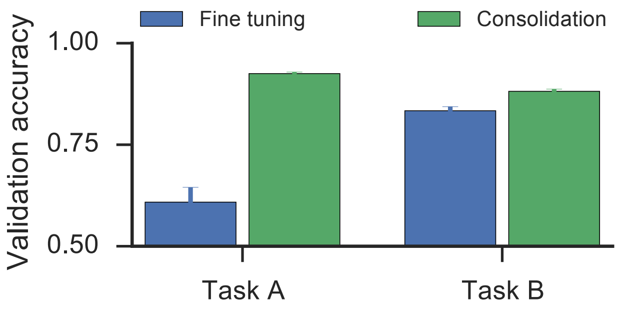

Figure 1. Classification accuracy for the split CIFAR-10 benchmark after being trained on Task A and B. Blue: Validation error, without consolidation ($c = 0$). Green: Validation error, with consolidation ($c = 0.1$). Note that chance-level in this benchmark is 0.2. Error bars correspond to SD (n=10).

C. Comparison of path integral approach to other metrics

Prior approaches toward measuring the sensitivity of parameters in a network have primarily focused on local metrics related to the curvature of the objective function at the final parameters (Martens, 2016). The Hessian is one possible metric, but it can be negative definite and computing even the diagonal adds additional overhead over standard backpropagation (Martens et al., 2012). An alterna-

[*]Equal contribution [1]Stanford University. Correspondence to: Friedemann Zenke <fzenke@stanford.edu>, Ben Poole <poole@cs.stanford.edu>.

Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

tive choice is the Fisher information (see for instance Kirkpatrick et al. (2017)):

$$\mathcal{F} = \mathbb{E}_{x\sim\mathcal{D},y\sim p_\theta(y|x)}\left[\left(\frac{\partial \log p_\theta(y|x)}{\partial\theta}\right)\left(\frac{\partial \log p_\theta(y|x)}{\partial\theta}\right)^T\right]$$

While the Fisher information has a number of desirable properties (Pascanu & Bengio, 2013), it requires computing gradients using labels sampled from the model distribution instead of the data distribution, and thus would require at least one additional backpropagation pass to compute online. For efficiency, the Fisher is often replaced with an approximation, the empirical Fisher (Martens, 2016), that uses labels sampled from the data distribution and can be computed directly from the gradient of the objective at the current parameters:

$$\begin{aligned}\bar{\mathcal{F}} &= \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\left(\frac{\partial \log p_\theta(y|x)}{\partial\theta}\right)\left(\frac{\partial \log p_\theta(y|x)}{\partial\theta}\right)^T\right]\\ &= \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[g(\theta)g(\theta)^T\right]\end{aligned}$$

The diagonal of the empirical Fisher yields a very similar formula to our local importance measure ω in Eq. 3 under gradient descent dynamics. However, the empirical Fisher is computed at a single parameter value $\boldsymbol{\theta}$ whereas the path integral is computed over a trajectory $\boldsymbol{\theta}(t)$. This yields an important difference in the behavior of these metrics: for a quadratic the empirical Fisher at the minimum will be 0 while the path integral will be proportional to the diagonal of the Hessian. Thus the path integral based approach yields an efficient algorithm with no additional gradients required that still recovers a meaningful estimate of the curvature.

References

Kirkpatrick, James, Pascanu, Razvan, Rabinowitz, Neil, Veness, Joel, Desjardins, Guillaume, Rusu, Andrei A., Milan, Kieran, Quan, John, Ramalho, Tiago, Grabska-Barwinska, Agnieszka, Hassabis, Demis, Clopath, Claudia, Kumaran, Dharshan, and Hadsell, Raia. Overcoming catastrophic forgetting in neural networks. *PNAS*, pp. 201611835, March 2017. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1611835114.

Martens, James. *Second-order optimization for neural networks*. PhD thesis, University of Toronto, 2016.

Martens, James, Sutskever, Ilya, and Swersky, Kevin. Estimating the hessian by back-propagating curvature. *arXiv preprint arXiv:1206.6464*, 2012.

Pascanu, Razvan and Bengio, Yoshua. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.