Bayesian Metaplasticity from Synaptic Uncertainty

Djohan Bonnet[1a,2], Tifenn Hirtzlin[1a], Tarcisius Januel[1a], Thomas Dalgaty[1b], Damien Querlioz[2], Elisa Vianello[1a]

[1] *Université Grenoble Alpes, CEA, LETI [a], LIST [b], Grenoble, France*
[2] *Université Paris-Saclay, CNRS, C2N, Palaiseau, France*
djohan.bonnet@cea.fr, damien.querlioz@c2n.upsaclay.fr, elisa.vianello@cea.fr

Abstract

Catastrophic forgetting remains a challenge for neural networks, especially in lifelong learning scenarios. In this study, we introduce MEtaplasticity from Synaptic Uncertainty (MESU), inspired by metaplasticity and Bayesian inference principles. MESU harnesses synaptic uncertainty to retain information over time, with its update rule closely approximating the diagonal Newton's method for synaptic updates. Through continual learning experiments on permuted MNIST tasks, we demonstrate MESU's remarkable capability to maintain learning performance across 100 tasks without the need of explicit task boundaries.

1 Introduction

Artificial Neural Networks (ANNs) have revolutionized various domains, but they suffer from catastrophic forgetting (Goodfellow et al., 2013) when trained on multiple tasks sequentially, limiting their applicability in lifelong learning scenarios. Many previous approaches to address this issue have focused on weight consolidation techniques (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2018). These methods identify important weights for previously learned tasks, typically using an approximation of the model's Hessian matrix. Simply, the Hessian diagonal component of each weight is used to protect it from further updates. While effective, these methods normally require clear task boundaries, which diminishes their real-world utility. In an embedded learning context, a neural network will continuously encounter new training data in an online fashion with no means of determining boundaries between tasks.

The human brain avoids catastrophic forgetting without task boundaries through mechanisms still not fully understood, but believed to involve synaptic metaplasticity (Fusi et al., 2005), a form of synaptic consolidation. Synapses adapt their learning rate continuously, not at the end of tasks. In ANNs, this type of metaplasticity has been shown empirically to reduce catastrophic forgetting (Laborieux et al., 2021; D'Agostino et al., 2023), but a theoretical understanding of these results is lacking. In parallel, recent work has also suggested that synaptic plasticity can be interpreted as Bayesian inference, where the synapse incorporates not just the synaptic weight but also a measure of its uncertainty (Aitchison et al., 2021). This raises intriguing questions: Could the concept of weight uncertainty value in this work be analogous to synaptic local learning rate in metaplasticity? If so, could a Bayesian approach be the key to enabling lifelong learning without catastrophic forgetting, mimicking the capabilities of the human brain?

In this paper, we propose a novel synaptic update rule based on Hessian-modulated learning and inspired by the principles of metaplasticity and Bayesian inference. We apply it to Bayesian neural networks (Blundell et al., 2015), which are ANNs incorporating uncertainty about the weights, reminiscent of the biological model of (Aitchison et al., 2021). Our experiments demonstrate that despite its fully spatially and temporally local nature, our metaplasticity rule matches the computation of the Hessian matrix. We also show experimentally its high performance in mitigating catastrophic forgetting, without the need for explicit task boundaries.

37th First Workshop on Machine Learning with New Compute Paradigms at NeurIPS 2023(MLNPCP 2023).

An existing method also leverages Bayesian neural networks for continual learning without task boundaries (Zeno et al., 2018), but it exhibits a marked deterioration in its ability to learn new tasks as more and more tasks are encountered. Our method, which shines new light on this work, in contrast maintains learning ability perpetually, achieving an operation that more closely resembles the capabilities of the human brain.

Our main contributions are as follows:

- The synaptic uncertainty in Bayesian Neural Networks can serve as a metaplasticity parameter, enabling continual learning without task boundaries, and which keeps on learning after a high number of tasks, aligning with the biological insights of Aitchison et al. (2021).
- Theoretically and experimentally, we demonstrate that the associated update rule allows for a precise approximation of the Hessian diagonal, providing synaptic updates equivalent to the diagonal approximation of Newton's method.
- Our algorithm matches the state-of-the-art of learning permuted MNIST tasks without knowing task boundaries, while maintaining the capacity to learn even after encountering 100 tasks.

2 Theoretical Results

In Bayesian neural networks (Blundell et al., 2015; Gal, 2016; Abdar et al., 2021), weights are typically represented as probability distributions defined by a mean-field Gaussian, $q(\omega|\theta)$, where θ encompasses both the mean (μ) and standard deviation (σ) values. Weight samples are expressed as $\omega = \mu + \epsilon \cdot \sigma$, with $\epsilon \sim \mathcal{N}(0, 1)$.

During the training phase, each weight is learned using the backpropagation algorithm. To calculate the gradients of the expectation of the negative log-likelihood, we employ the Bayes by Backprop method. Consequently, the cost function is defined as $\mathcal{C} = \mathbb{E}_{q(\omega|\theta)}[\mathcal{L}(\omega)]$, where $\mathcal{L}$ is the negative log-likelihood, and the gradients are computed as

$$\frac{\partial \mathcal{C}}{\partial \mu} = \mathbb{E}_\epsilon \left[\frac{\partial \mathcal{L}(\omega)}{\partial \omega} \right] \qquad \frac{\partial \mathcal{C}}{\partial \sigma} = \mathbb{E}_\epsilon \left[\frac{\partial \mathcal{L}(\omega)}{\partial \omega} \times \epsilon \right]. \tag{1}$$

Using a learning rate γ, σ and μ at the update iteration $n + 1$ are given by

$$\mu_{n+1} = \mu_n - \gamma \times \frac{\partial \mathcal{C}}{\partial \mu_n} \qquad \sigma_{n+1} = \sigma_n - \gamma \times \frac{\partial \mathcal{C}}{\partial \sigma_n}. \tag{2}$$

Like standard ANNs trained by the backpropagation algorithm, Bayesian neural networks trained by Bayes by Backprop are prone to catastrophic forgetting. To address the challenge of continual learning in Bayesian neural networks, our goal is to construct a metaplasticity function based on the continual computation of the Hessian diagonal of the negative log-likelihood. This Hessian diagonal serves to quantify the "importance" associated with each synapse. Synapses of greater importance exhibit a large contribution to the Hessian diagonal, requiring a reduction of their local learning rate. However, while the utility of the Hessian is clear, it is costly to evaluate, and its continual evaluation is usually not feasible. Following the biology-inspired principle of metaplasticity (Fusi et al., 2005; Laborieux et al., 2021; Jedlicka et al., 2022), in our approach, the Hessian diagonal is computed naturally using simple equations.

To understand our approach, we first highlight a connection between synaptic standard deviation (σ) and the Hessian diagonal. We demonstrate Lemma 1 in Appendix A, which shows that when a component of the Hessian matrix has a high positive value, it results in a significant gradient with respect to the corresponding σ, leading to a decrease in σ according to Eq. (2). This observation suggests that synapses with greater importance for the previously learned tasks tend to have smaller σ values.

Lemma 1 (Hessian diagonal via First-Order Derivative in Bayesian Neural Networks)**.** *Consider a mean-field Gaussian, $q(\omega|\theta)$, describing a Bayesian neural network, where $\theta = (\mu, \sigma)$, and weight samples are defined as $\omega = \mu + \epsilon \cdot \sigma$, with $\epsilon \sim \mathcal{N}(0, 1)$. Let $\mathcal{L}$ be the negative log-likelihood and*

$\mathcal{C}$ the expectation of $\mathcal{L}$, defined as $\mathcal{C} = \mathbb{E}_{q(\omega|\theta)}[\mathcal{L}(\omega)]$. The diagonal of the Hessian matrix of the expectation with respect to μ is given by: $H_D(\mu) = \frac{1}{\sigma}\frac{\partial \mathcal{C}}{\partial \sigma}$.

We now propose to modify the updates on both σ and μ (Eq. (2)) to achieve an update rule that we will show to be equivalent to the diagonal approximation of Newton's method (Becker et al., 1988).

$$\boxed{\mu_{n+1} = \mu_n - \sigma_n^2 \times \frac{\partial \mathcal{C}}{\partial \mu_n} \qquad \sigma_{n+1} = \sigma_n - \sigma_n^2 \times \frac{\partial \mathcal{C}}{\partial \sigma_n} + \frac{\sigma_n(\sigma_{prior}^2 - \sigma_n^2)}{\sigma_{res}^2}}. \tag{3}$$

where σ_n^2 is the metaplasticity function, σ_{prior} and σ_{res} are two positive hyperparameters, with $\sigma_{prior}^2 \ll \sigma_{res}^2$. σ_n is attracted towards σ_{prior}, which can therefore be interpreted as a prior belief over σ_n (MacKay, 1992). σ_{res} regulates the strength of the link between σ_n and σ_{prior}.

We call these update rules "MEtaplasticity from Synaptic Uncertainty" (MESU). When learning a new task $n+1$, MESU preserves the performance of the previously learned task n by constraining the mean-field Gaussian $q(\omega|\theta_{n+1})$ to remain within a region of overlap with $q(\omega|\theta_n)$, as illustrated in Fig. 1. This constraint is implemented through the metaplasticity function σ_n^2 and aligns with parallel observations from computational neuroscience of Aitchison et al. (2021).

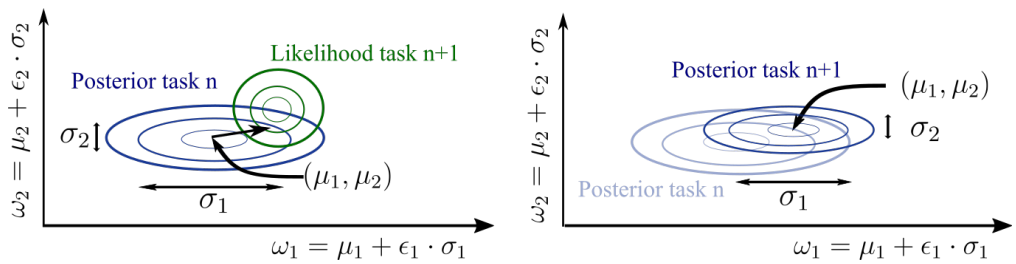

Figure 1: Qualitative illustration of the update rule. Contour plot of the density of the mean-field Gaussians $q(\omega|\theta_n)$ and $q(\omega|\theta_{n+1})$ for the new task $n+1$ and the previously learned task n (blue). Contour plot of the Likelihood of the incoming task number $n+1$ (green).

The Locally Constant Curvature To justify MESU, we make a necessary assumption that, locally, the curvature of the loss function is constant and positive. This implies that for each sample ω, $\frac{\partial^2 \mathcal{L}(\omega)}{\partial \omega^2}$ is a positive constant, which we call $\frac{1}{\sigma_L^2}$. This assumption leads to two important consequences. First, the Hessian diagonal with respect to σ is equivalent to the Hessian diagonal with respect to μ:

$$H_D(\mu) = \mathbb{E}_\epsilon\left[\frac{\partial^2 \mathcal{L}(\omega)}{\partial \omega^2}\right] = \frac{1}{\sigma_L^2} \tag{4}$$

$$H_D(\sigma) = \mathbb{E}_\epsilon\left[\frac{\partial^2 \mathcal{L}(\omega)}{\partial \omega^2}\right] \times \epsilon^2 = \frac{\partial^2 \mathcal{L}(\omega)}{\partial \omega^2} \times \mathbb{E}_\epsilon[\epsilon^2] = \frac{1}{\sigma_L^2}. \tag{5}$$

Second, the update over σ becomes a recurrent sequence that can be analyzed. Using the lemma, Eq. (3), and Eq. (4), we obtain:

$$\sigma_{n+1} = \sigma_n(1 - \frac{\sigma_n^2}{\sigma_L^2}) + \frac{\sigma_n(\sigma_{prior}^2 - \sigma_n^2)}{\sigma_{res}^2}. \tag{6}$$

A convergence analysis of the sequence result leads to our theorem (see Appendix C for full proof).

Theorem 1. *[Hessian Diagonal via Synaptic Uncertainty in Bayesian Neural Networks] Let $q(\omega|\theta)$ represent the mean-field Gaussian describing the Bayesian neural network, where $\theta = (\mu, \sigma)$, and weight samples are defined as $\omega = \mu + \epsilon \cdot \sigma$, with $\epsilon \sim \mathcal{N}(0, 1)$. Consider $\mathcal{L}$ as the negative log-likelihood and $\mathcal{C}$ as the expectation of $\mathcal{L}$, defined as $\mathcal{C} = \mathbb{E}_{q(\omega|\theta)}[\mathcal{L}(\omega)]$. Let $\sigma_{n+1} = \sigma_n - \sigma_n^2 \times \frac{\partial \mathcal{C}}{\partial \sigma_n} +$*

$\frac{\sigma_n(\sigma_{prior}^2-\sigma_n^2)}{\sigma_{res}^2}$ *be the update rule over* σ*. If the loss curvature can be expressed as* $\frac{1}{\sigma_L^2}$*,* $\sigma_0 < \sigma_L$ *and,* $\frac{\sigma_{prior}^2}{\sigma_{res}^2} \ll 1$*, then the diagonal elements of the Hessian matrix with respect to* μ *and* σ *are given by:*

$$H_D(\mu) = H_D(\sigma) = \lim_{n\to+\infty} \left[\frac{\sigma_{prior}^2}{\sigma_{res}^2 \sigma_n^2} - \frac{1}{\sigma_{res}^2} \right]. \tag{7}$$

According to Theorem 1, for sufficiently large n, the metaplasticity function σ_n^2 can be rewritten as:

$$\boxed{\lim_{n\to+\infty} \sigma_n^2 = \frac{\sigma_{prior}^2}{\sigma_{res}^2} \times \frac{1}{H_D(\mu) + \frac{1}{\sigma_{res}^2}}}. \tag{8}$$

By substituting Eq. (8) into Eq. (3), the update on μ becomes proportional to the inverse of the Hessian diagonal. MESU is therefore equivalent to the diagonal approximation of Newton's method with a learning rate of $\frac{\sigma_{\text{prior}}^2}{\sigma_{\text{res}}^2}$ and a residual of $\frac{1}{\sigma_{res}^2}$. This constitutes the main theoretical results of the paper. The residual term provides a safeguard against a null Hessian diagonal component.

Importance of the regularization term The term $\frac{\sigma_n(\sigma_{\text{prior}}^2-\sigma_n^2)}{\sigma_{\text{res}}^2}$ introduced in the σ_{n+1} update in Eq. (3) constitutes the primary difference between MESU and the Bayesian Gradient Descent (BGD) method of Zeno et al. (2018). Under the condition $\frac{\partial \mathcal{C}}{\partial \sigma_n} \times \sigma_n \ll 1$ and with a scaling factor of 0.5 to the σ update, Eq. (3) aligns with BGD when this term is omitted. This addition is crucial for two reasons. First, it prevents standard deviations from reaching zero, as shown in Eq. (10). Second, it prevents large changes in the mean value while maintaining a constant standard deviation. Similarly, the synaptic metaplasticity model introduced by Fusi et al. (2005) inhibits any synaptic changes in value while maintaining a constant metaplastic value. To further explore the relationship between MESU and BGD, in Appendix C, Theorem 2, we express the MESU update for σ without the regularization term as a recurrent sequence and show that:

$$H_D(\mu) = H_D(\sigma) = \lim_{n\to+\infty} \frac{1}{2n\sigma_n^2}. \tag{9}$$

This result allows us to derive the analog of a metaplasticity function for the BGD method:

$$\sigma_n^2 = \frac{1}{2nH_D(\mu)} \qquad \lim_{n\to+\infty} \frac{1}{2nH_D(\mu)} = 0. \tag{10}$$

Consequently, the BGD method is also equivalent to the Newton method, but, in that case, when the number of iterations n is sufficiently large, the metaplasticity function, and therefore, the update on μ, approach zero. This results in a linear decrease in plasticity with increasing number of iterations. In the following section, we realize that this difference between MESU and BGD has important consequences when learning a large number of tasks.

3 Experimental Results

We now evaluate our proposed MESU update on a Bayesian neural network trained on MNIST and its suitability for continual learning through the permuted MNIST task (LeCun, 1998; Goodfellow et al., 2013). We compare MESU to other approaches which rely on Hessian matrix approximation, and to BGD (Zeno et al., 2018). Appendix D provides full implementation details.

Validation of theoretical results We start by presenting various empirical results on the MNIST dataset. We employ a small fully-connected neural network with a single hidden layer of 100 units trained with MESU (Eq. (3)). To quantify the capability of the MESU trained model to approximate the Hessian, we compute the true Hessian diagonal with respect to both σ and μ using the PyTorch autograd function, based on a snapshot of the model synapses after 100,000 updates. First, we investigate our critical assumption of locally constant curvature (Eq.(4)), which predicted that the Hessian

of the mean and standard deviation would be equivalent. In Fig. 2a, we plot, as a point per synapse, the true Hessian diagonal with respect to σ and μ where we observe that they are indeed equivalent. Fig. 2b shows that the Hessian diagonal with respect to μ is accurately approximated by Eq. (7). This result substantiates Theorem 1, the most important theoretical result of our work and shows that MESU is equivalent to Newton's method. Fig. 2c shows empirical support for Theorem 2: a Bayesian neural network trained using BGD is also equivalent to Newton's method to a large extent, but only when Hessian diagonal components are above 5×10^{-4}. For BGD, due to the absence of the regularization term present in MESU, σ_n value remains constrained to the initialization value, $\sigma_{init} = 0.1$, which impedes the reduction of Hessian diagonal components.

In Fig. 2d, we study the temporal evolution of the Hessian approximation of these two methods. We also evaluate MESU and BGD against two state-of-the-art Hessian approximation-based methods, elastic weight consolidation (EWC) (Kirkpatrick et al., 2017) and synaptic intelligence (SI) (Zenke et al., 2017). In MESU and BGD the quality of the Hessian approximation improves with an increasing number of iterations, n. For a lower n, the approximation is worse, as the hypothesis of locally constant curvature is not yet satisfied. However, as training progresses and the model converges to a region where the local curvature becomes constant, the approximation greatly improves. Our theoretical result holds as n approaches infinity. At n=100,000 iterations, the Hessian approximation accuracy provided by MESU exceeds not only EWC and SI, but also that of BGD.

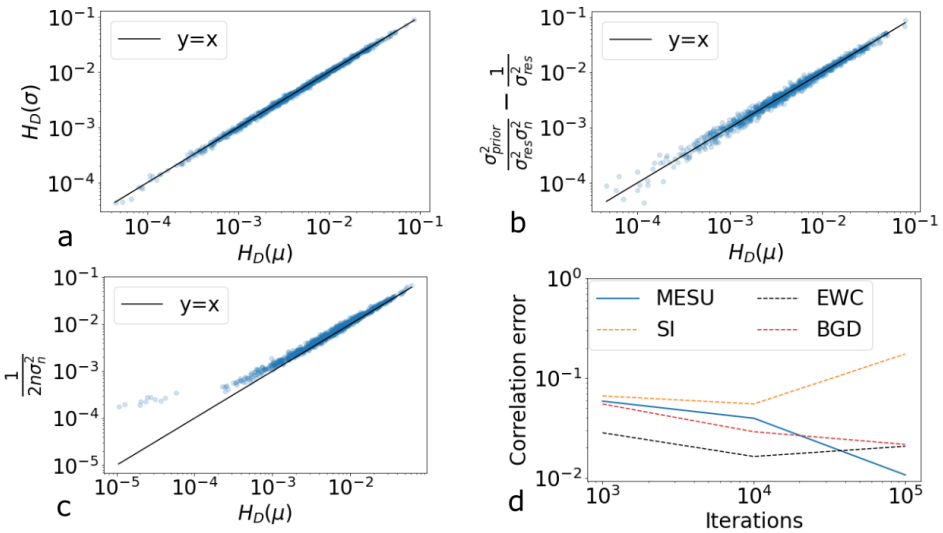

Figure 2: Verification of theoretical results. a Hessian diagonal with respect to μ vs. the Hessian diagonal with respect to σ after n=100,000 iterations. **b** Approximation of the Hessian diagonal by MESU (Eq. (7)) vs. actual Hessian diagonal. **c** Approximation of the Hessian diagonal by MESU without regularization term vs. actual Hessian diagonal. **d** Correlation between the Hessian actual matrix and its approximation for EWC, SI, and MESU as a function of iteration updates.

Vanishing plasticity One key advantage of MESU over BGD should be that, as the number of iterations n increases, the σ parameters do not tend to zero - an effect we refer to as *vanishing plasticity*: with MESU, the network should remain plastic (i.e., it can continually learn new tasks) due to the regularization term. To test this, we trained the neural network on 100 permutations of MNIST (Van de Ven and Tolias, 2019) using both MESU and BGD. The test accuracy in each of 100 tasks is evaluated after each epoch. Each permutation is learned over 20 epochs. Both models operate without any prior knowledge of task boundaries, complicating the preservation of previous knowledge. As shown in Fig. 3a, the tasks evolve continuously, and the MESU method mitigates catastrophic forgetting. Fig. 3b monitors the range of σ values assumed by the models synapses as the learning process progresses for both methods. In the MESU method, the mean value of σ remains relatively constant throughout the entire training process. In contrast, for BGD, σ decreases as the number of epochs increases. After 2000 epochs, corresponding to 100 tasks, the low σ values will impede the network from learning new tasks effectively, indicating the vanishing plasticity problem. This difference has important consequences on the learning process. We assessed continual learning

at two key stages: the initial phase (the first ten tasks, Fig. 3c) and the later phase (the last ten tasks, Fig. 3d). In the initial phase, BGD demonstrates a slight advantage, displaying less forgetting of the initial ten tasks. However, after 100 tasks, BGD experiences a significant drop in accuracy of around 6%, while MESU retains its ability to learn, almost equivalently to the initial phase.

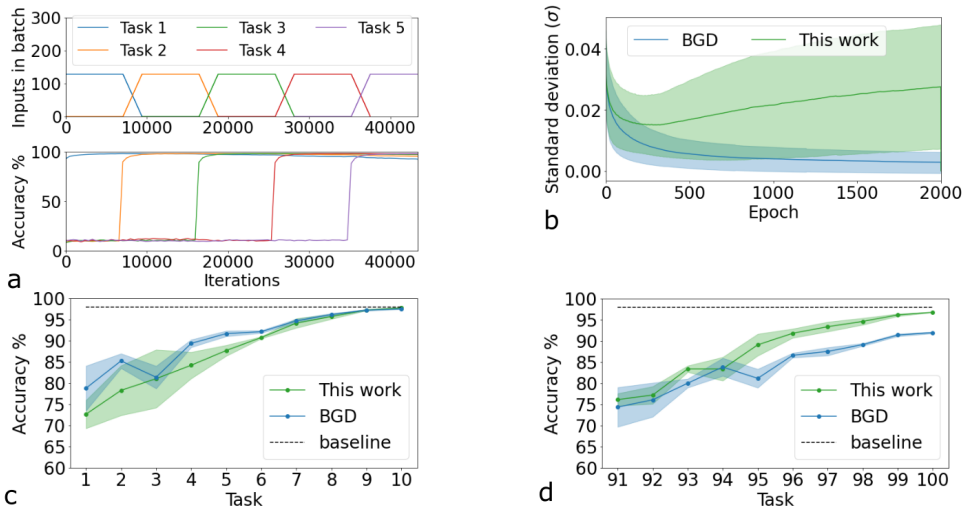

Figure 3: Benchmark between BGD and MESU. **a** Evolution of the number of input belonging to each task (different permutation of MNIST) present in a training batch as a function of the number of iterations, along with the evolution of MESU test accuracy corresponding to each task. **b** Evolution of the mean value of the synaptic uncertainty with respect to the number of epochs for BGD and MESU. **c,d** Test accuracy for the last ten tasks learned after learning ten tasks (**c**) and 100 tasks (**d**) for BGD and MESU, averaged over 3 trials. Error bars represent one standard deviation.

4 Discussion

This paper presents three contributions: (1) the establishment of a link between synaptic uncertainty and the Hessian diagonal, (2) the introduction of a novel update rule, MEtaplasticity from Synaptic Uncertainty (MESU), which allows continuous learning without vanishing plasticity, and (3) a theoretical and empirical validation of these concepts. Our work also offers a novel interpretation of the previously proposed Bayesian Gradient Descent (BGD), enriching its theoretical underpinnings.

Our method is not alone in attempting continual learning without task boundaries. Zeno et al. (2018) proposed task-boundary-less versions of Online Elastic Weight Consolidation (Schwarz et al., 2018) and Memory-Aware Synapses (Aljundi et al., 2018). While these methods show promise, previous work showed that BGD outperforms them in the absence of task boundaries (Zeno et al., 2018). However, all these techniques, including BGD, deteriorate in performance as the number of tasks increases. Our method not only largely matches BGD's performance for a limited number of initial tasks, but also maintains its learning capability over a massively extended range of tasks where BGD suffers from vanishing plasticity.

Replay-based methods are another avenue for addressing catastrophic forgetting. They require sophisticated sample selection when task boundaries are absent (Aljundi et al., 2019). Still, an exciting future direction would be to investigate the potential synergies (Nguyen et al., 2017) between replay-based methods and MESU to achieve even better performance.

Our method inherits the computational challenges associated with Bayesian Neural Networks, primarily the expensive sampling process. While this is a limitation, recent work on hardware implementations of Bayesian Neural Networks, using e.g., memristors, suggests a promising avenue (Dalgaty et al., 2021; Lin et al., 2023; Bonnet et al., 2023). Our work shows a unique convergence between Hessian computation, biological insights, and Bayesian principles, which is not only informative, but offers a deeper understanding of how to solve the stability-plasticity dilemma (Mermillod et al., 2013) in long-term continual learning scenarios.

References

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.

Laurence Aitchison, Jannes Jegminat, Jorge Aurelio Menendez, Jean-Pascal Pfister, Alexandre Pouget, and Peter E Latham. Synaptic plasticity as bayesian inference. *Nature neuroscience*, 24(4):565–571, 2021.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

Sue Becker, Yann Le Cun, et al. Improving the convergence of back-propagation learning with second order methods. In *Proceedings of the 1988 connectionist models summer school*, pages 29–37, 1988.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

Djohan Bonnet, Tifenn Hirtzlin, Atreya Majumdar, Thomas Dalgaty, Eduardo Esmanhotto, Valentina Meli, Niccolò Castellani, Simon Martin, Jean-Francois Nodin, Guillaume Bourgeois, et al. Bringing uncertainty quantification to the extreme-edge with memristor-based bayesian neural networks. 2023.

Simone D'Agostino, Filippo Moro, Tifenn Hirtzlin, Julien Arcamone, Niccolò Castellani, Damien Querlioz, Melika Payvand, and Elisa Vianello. Synaptic metaplasticity with multi-level memristive devices. *arXiv preprint arXiv:2306.12142*, 2023.

Thomas Dalgaty, Eduardo Esmanhotto, Niccolo Castellani, Damien Querlioz, and Elisa Vianello. Ex situ transfer of bayesian neural networks to resistive memory-based inference hardware. *Advanced Intelligent Systems*, 3(8):2000103, 2021.

Stefano Fusi, Patrick J Drew, and Larry F Abbott. Cascade models of synaptically stored memories. *Neuron*, 45(4):599–611, 2005.

Yarin Gal. Uncertainty in deep learning. *PhD thesis, University of Cambridge*, 2016.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Peter Jedlicka, Matus Tomko, Anthony Robins, and Wickliffe C Abraham. Contributions by metaplasticity to solving the catastrophic forgetting problem. *Trends in Neurosciences*, 45(9):656–666, 2022.

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

Axel Laborieux, Maxence Ernoult, Tifenn Hirtzlin, and Damien Querlioz. Synaptic metaplasticity in binarized neural networks. *Nature communications*, 12(1):2549, 2021.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yudeng Lin, Qingtian Zhang, Bin Gao, Jianshi Tang, Peng Yao, Chongxuan Li, Shiyu Huang, Zhengwu Liu, Ying Zhou, Yuyi Liu, et al. Uncertainty quantification via a memristor bayesian deep neural network for risk-sensitive reinforcement learning. *Nature Machine Intelligence*, pages 1–10, 2023.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013.

Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, pages 4528–4537. PMLR, 2018.

Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

Chen Zeno, Itay Golan, Elad Hoffer, and Daniel Soudry. Task agnostic continual learning using online variational bayes. *arXiv preprint arXiv:1803.10123*, 2018.

Acknowledgements

This work was supported by European Research Council consolidator grant DIVERSE (reference: 101043854) and by EIC Pathfinder METASPIN grant (reference: 101098651). In addition, we thank L. Hutin, A. Laborieux, for discussing various aspects of the article.

Appendices

A: Lemma

Lemma 1 (Hessian diagonal via First-Order Derivative in Bayesian Neural Networks)**.** *Consider a mean-field Gaussian, $q(\omega|\theta)$, describing a Bayesian neural network, where $\theta = (\mu, \sigma)$, and weight samples are defined as $\omega = \mu + \epsilon \cdot \sigma$, with $\epsilon \sim \mathcal{N}(0, 1)$. Let $\mathcal{L}$ be the negative log-likelihood and $\mathcal{C}$ the expectation of $\mathcal{L}$, defined as $\mathcal{C} = \mathbb{E}_{q(\omega|\theta)}[\mathcal{L}(\omega)]$. The diagonal of the Hessian matrix of the expectation with respect to μ is given by: $H_D(\mu) = \frac{1}{\sigma}\frac{\partial \mathcal{C}}{\partial \sigma}$.*

Proof. By definition, the diagonal elements of the Hessian matrix with respect to μ are: $H_D(\mu) = \frac{\partial^2 \mathcal{C}}{\partial \mu^2}$. Given that: $\frac{\partial \mathcal{C}}{\partial \mu} = \mathbb{E}_\epsilon \left[\frac{\partial \mathcal{L}(\omega)}{\partial \omega}\right]$, we deduce: $H_D(\mu) = \mathbb{E}_\epsilon \left[\frac{\partial^2 \mathcal{L}(\omega)}{\partial \omega^2}\right]$.

Similarly, we have: $\frac{\partial \mathcal{C}}{\partial \sigma} = \mathbb{E}_\epsilon \left[\frac{\partial \mathcal{L}(\omega)}{\partial \omega} \times \epsilon\right]$.

Using Stein's lemma, we obtain the following relation: $\mathbb{E}_\omega \left[\frac{\partial \mathcal{L}(\omega)}{\partial \omega} \times (\omega - \mu)\right] = \sigma^2 \mathbb{E}_\omega \left[\frac{\partial^2 \mathcal{L}(\omega)}{\partial \omega^2}\right]$. Therefore, $\mathbb{E}_\epsilon \left[\frac{\partial \mathcal{L}(\omega)}{\partial \omega} \times \epsilon\right] = \sigma \mathbb{E}_\epsilon \left[\frac{\partial^2 \mathcal{L}(\omega)}{\partial \omega^2}\right]$.

We recognize the left term as $\frac{\partial \mathcal{C}}{\partial \sigma}$ and conclude:

$$\boxed{H_D(\mu) = \frac{1}{\sigma}\frac{\partial \mathcal{C}}{\partial \sigma}}. \tag{11}$$

□

B: Sequences Analysis

Proposition 1 (Study of a sequence convergence for Theorem 1)**.** *Consider the sequence* $\alpha_{n+1} = \frac{\alpha_n(1-\alpha_n^2)}{1-\alpha_{lim}^2}$*, with* $0 < \alpha_0 < 1$ *and* $0 < \alpha_{lim} < \frac{1}{\sqrt{3}}$*. Then, the sequence on* (α_n) *converges to:*

$$\lim_{n\to+\infty} \alpha_n = \alpha_{lim}. \tag{12}$$

Proof. To demonstrate the convergence of this sequence we examine the two possible situations: $\alpha_0 < \alpha_{lim}$ and $\alpha_0 > \alpha_{lim}$.

(i): Firstly, we consider $\alpha_0 < \alpha_{lim}$.

By induction we establish that this upper bound stands $\forall n \in \mathbb{N},\ \alpha_n < \alpha_{lim}$.

Base Case (Initialization): For n=0

This holds true by the given assumptions.

Inductive Step: Assume for some arbitrary $n \in \mathbb{N}$ that : $\alpha_n < \alpha_{lim}$.

$$\alpha_{n+1} - \alpha_{lim} = \frac{\alpha_n(1-\alpha_n^2) - \alpha_{lim}(1-\alpha_{lim}^2)}{1-\alpha_{lim}^2}. \tag{13}$$

We define $f(x) = x(1-x^2)$. A simple analysis of the function gives f increasing over the interval $[0, \frac{1}{\sqrt{3}}]$ and decreasing over the interval $[\frac{1}{\sqrt{3}}, 1]$ reaching its maximum value of $\frac{2}{3\sqrt{3}}$ at $\frac{1}{\sqrt{3}}$. The sequence is rewritten:

$$\alpha_{n+1} - \alpha_{lim} = \frac{f(\alpha_n) - f(\alpha_{lim})}{1-\alpha_{lim}^2}. \tag{14}$$

Given that the function f is increasing over the interval $[0, \frac{1}{\sqrt{3}}]$ and that $0 < \alpha_n < \alpha_{lim} < \frac{1}{\sqrt{3}}$, we can deduce that $f(\alpha_n) - f(\alpha_{lim}) < 0$. Consequently: $\alpha_{n+1} < \alpha_{lim}$.

Therefore,

$$\forall n \in \mathbb{N},\ \alpha_n < \alpha_{lim}. \tag{15}$$

According to the definition of the sequence $\alpha_{n+1} > \alpha_n$, hence the sequence (α_n) is monotonically increasing. The sequence (α_n) is both increasing and upper bounded by α_{lim}, (α_n) therefore, the function must converge. We denote the limit of this convergence as l.

When convergence is achieved the sequence becomes: $l = \frac{l(1-l^2)}{1-\alpha_{lim}^2}$ and we can deduce: $l = \alpha_{lim}$.

Finally:

$$0 < \alpha_0 < \alpha_{lim} < \frac{1}{\sqrt{3}} \implies \lim_{n\to+\infty} \alpha_n = \alpha_{lim}. \tag{16}$$

(ii): Secondly, we consider $\alpha_{lim} < \alpha_0$.

If there exists an n_0 such that $\alpha_{n_0} < \alpha_{lim}$, then we can apply result of Eq. (16) of the previous analysis. If no such n_0 exists, $\forall n \in \mathbb{N}$, $\alpha_{lim} < \alpha_n$, we can immediately infer from the sequence's definition that $\alpha_{n+1} < \alpha_n$. The sequence (α_n) is lower bounded by α_{lim}, which implies that the sequence (α_n) is monotonically decreasing. Therefore the sequence (α_n) converges and the limit is α_{lim}, concluding the proof for any initial values satisfying $0 < \alpha_0 < 1$ and $0 < \alpha_{lim} < \frac{1}{\sqrt{3}}$:

$$\boxed{\lim_{n\to+\infty} \alpha_n = \alpha_{lim}}. \tag{17}$$

□

Proposition 2 (Study of a sequence's asymptotic behavior for Theorem 2)**.** *Consider the sequence* $\alpha_{n+1} = \alpha_n(1-\alpha_n^2)$ *where* $0 < \alpha_0 < 1$*. Then, the sequence on* (α_n) *converge to zero, and :*

$$\lim_{n\to+\infty} \alpha_n\sqrt{2n} = 1. \tag{18}$$

(i): By induction, we establish that: $\forall n \in \mathbb{N}^*$, $\alpha_n \leq \frac{1}{\sqrt{2n}}$.

Base Case (Initialization): n=1,2

From proposition 1 we establish that $f(x) = x(1-x^2) \leq \frac{2}{3\sqrt{3}}$.

Thus, $\alpha_1 \leq \frac{2}{3\sqrt{3}} \leq \frac{1}{\sqrt{2}}$ and $\alpha_2 \leq \frac{2}{3\sqrt{3}} \leq \frac{1}{\sqrt{4}}$, confirming the base case.

Inductive Step: We assume the proposition holds for $n \geq 2 \in \mathbb{N}$, i.e., $\alpha_n \leq \frac{1}{\sqrt{2n}}$.

As f is monotonically increasing on $[0, \frac{1}{\sqrt{3}}]$, and from the inductive hypothesis $f(\alpha_n) \leq f(\frac{1}{\sqrt{2n}})$ provided:

$$\alpha_{n+1} \leq \frac{1}{\sqrt{2n}}(1-\frac{1}{2n}). \tag{19}$$

Using Taylor's expansion we have the following expression :

$$\frac{1}{\sqrt{2n+2}} = \frac{1}{\sqrt{2n}}(1-\frac{1}{2n}+\frac{3}{8n^2}+R_3(\frac{1}{n})). \tag{20}$$

Where R_3 represents the remainder of the Taylor series and Taylor's inequality gives $\forall n \geq 2$, $|R_3(\frac{1}{n})| < \frac{3}{8n^2}$. Eq.(19) and Eq.(20) lead to $\alpha_{n+1} \leq \frac{1}{\sqrt{2n+2}}$

$$\forall n \in \mathbb{N}^*, \alpha_n \leq \frac{1}{\sqrt{2n}}. \tag{21}$$

(ii): We consider a sequence (V_n) defined as: $V_n = \frac{n}{2} - n^2\alpha_n^2$ and demonstrate that the sequence (V_n) is monotonically increasing. Taking the difference between successive terms, we have:

$$V_{n+1} - V_n = \frac{1}{2} - (n+1)^2\alpha_{n+1}^2 + n^2\alpha_n^2. \tag{22}$$

Using the result from (i) we ascertain that: $(n+1)^2\alpha_{n+1}^2 \leq \frac{1}{2}$.

From which we infer $V_{n+1} - V_n \geq n^2\alpha_n^2 \geq 0$

Involving that the sequence (Vn) is increasing.There is two possibilities: First possibility, (V_n) converges and the limit is:

$$\lim_{n\to+\infty} V_n = c \in \mathbb{R}. \tag{23}$$

In consequence :

$$\lim_{n\to+\infty} \frac{V_n}{n} = \lim_{n\to+\infty} n(\frac{1}{2n} - \alpha_n^2) = 0. \tag{24}$$

Concluding the proof when (V_n) converge:

$$\boxed{\lim_{n\to+\infty} \alpha_n\sqrt{2n} = 1}. \tag{25}$$

Second possibility:

$$\lim_{n\to+\infty} V_n = +\infty. \tag{26}$$

(iii): We consider the sequence (P_n) defined by: $P_n = \alpha_n\sqrt{2n}$.

We demonstrate that there is a given $n_0 \in \mathbb{N}^*$ such that $\forall n \geq n_0$, $P_n \leq P_{n+1}$. Involving (P_n) increasing beyond n_0.

Looking at the ratio between successive terms, we find:

$$\frac{P_{n+1}}{P_n} = \frac{\alpha_n(1-\alpha_n^2)\sqrt{2n+2}}{\alpha_n\sqrt{2n}} = (1-\alpha_n^2)\sqrt{1+\frac{1}{n}}. \tag{27}$$

Further expanding, we have:

$$\frac{P_{n+1}}{P_n} = 1 - \alpha_n^2 + \frac{1}{2n} + O(\frac{1}{n^2}). \tag{28}$$

Using the results from (i), we ascertain $\frac{1}{2n} - \alpha_n^2 \geq 0$. Additionally, by invoking (ii), it becomes clear that as n grows, the term $O(\frac{1}{n^2})$ become negligible relative to $(\frac{1}{2n} - \alpha_n^2)$. Consequently, we can deduce that there exists a particular $n_0 \in \mathbb{N}^*$ for which all $n \geq n_0$, $\frac{P_{n+1}}{P_n} \geq 1$.

Concluding to:

$$\exists n_0 \in \mathbb{N}^*, \forall n \geq n_0, (P_n) \nearrow . \tag{29}$$

From (i) we know that $(P_n) \leq 1$ and from (iii) we know that (P_n) is monotically increasing and so (P_n) converge:

$$\lim_{n\to+\infty} P_n = c. \tag{30}$$

(iv) We consider $(log(P_n))$ demonstrating the convergence of the series $\sum(\frac{1}{2n} - \alpha_n^2)$.

First, let us rewrite (α_n).

$$\alpha_{n+2} = \alpha_{n+1}(1-\alpha_{n+1}^2) = \alpha_n(1-\alpha_n^2)(1-\alpha_{n+1}^2). \tag{31}$$

From the above recursion relation, we can inductively deduce the general form:

$$\alpha_n = \prod_{k=1}^{n-1}(1-\alpha_k^2)\alpha_1. \tag{32}$$

This allows us to express $(log(P_n))$ as:

$$log(P_n) = \sum_{k=1}^{n-1} log((1-\alpha_k^2)) + log(\alpha_1) + log(\sqrt{2n}), \tag{33}$$

which, in the limit as n approaches infinity, is asymptotic to:

$$log(P_n) \underset{n\to+\infty}{\sim} \sum_{k=1}^{n-1} log((1-\alpha_k^2)) + \frac{1}{2}log(n). \tag{34}$$

From a prior result (i), we have:

$$log((1-\alpha_k^2)) = -\alpha_k^2 + O(\frac{1}{n^2}). \tag{35}$$

Invoking the properties of the Harmonic series, we can further deduce:

$$log(P_n) \underset{n\to+\infty}{\sim} \sum_{k=1}^{n-1} -\alpha_k^2 + \frac{1}{2}\sum_{k=1}^{n-1}\frac{1}{k}. \tag{36}$$

Given Equation (30):

$$\lim_{n\to+\infty} log(P_n) = log(c). \tag{37}$$

In consequence the series $\sum(\frac{1}{2n} - \alpha_n^2)$ converge.

Proposition 2 is demonstrated based on the previous arguments .Based on previous results, namely (i) and (iii), we know that $(\frac{1}{2n} - \alpha_n^2) \geq 0$ and the sequence $(\frac{1}{2n} - \alpha_n^2)$ is monotonically decreasing.

We consider the two sequences (u_n) and (v_n) such that $u_n = \frac{1}{2n} - \alpha_n^2$ and $v_n = n(u_n - u_{n+1})$. By rearranging terms, the sum of the first n terms of v_k can be expressed as:

$$\sum_{k=0}^{n} v_k = \sum_{k=1}^{n} ku_k - \sum_{k=1}^{n+1}(k-1)u_k, \tag{38}$$

which gives:

$$\sum_{k=0}^{n} u_k = \sum_{k=0}^{n} v_k + nu_{n+1}. \tag{39}$$

Given that $\sum u_k$ converges in (iv), it follows that $(\sum_{k=0}^{n} v_k + nu_{n+1})$ converges too. Since $u_n \geq 0$ and the sequence (u_n) is monotonically decreasing, $v_n \geq 0$. Hence, both of the series $\sum v_k$ and nu_{n+1} converge.

Considering the limit of the product nu_n, we have:

$$\lim_{n\to+\infty} nu_n = l. \tag{40}$$

If $l > 0$, we have $u_n \underset{n\to+\infty}{\sim} \frac{l}{n}$. However, this would lead to the divergence of $\sum u_k$ and consequently we deduce that $l = 0$ and therefore:

$$\lim_{n\to+\infty} nu_n = 0, \tag{41}$$

from which we can deduce:

$$\lim_{n\to+\infty} \frac{V_n}{n} = \lim_{n\to+\infty} n(\frac{1}{2n} - \alpha_n^2) = 0. \tag{42}$$

Concluding the proof when $V_n \to \infty$.:

$$\boxed{\lim_{n\to+\infty} \alpha_n\sqrt{2n} = 1} \tag{43}$$

C: Theorems demonstration

Theorem 1. *[Hessian Diagonal from Synaptic Uncertainty in Bayesian Neural Networks] Let $q(\omega|\theta)$ represent the mean-field Gaussian describing the Bayesian neural network, where $\theta = (\mu, \sigma)$, and weight samples are defined as $\omega = \mu + \epsilon \cdot \sigma$, with $\epsilon \sim \mathcal{N}(0, 1)$. Consider $\mathcal{L}$ as the negative log-likelihood and $\mathcal{C}$ as the expectation of $\mathcal{L}$, defined as $\mathcal{C} = \mathbb{E}_{q(\omega|\theta)}[\mathcal{L}(\omega)]$. Let $\sigma_{n+1} = \sigma_n - \sigma_n^2 \times \frac{\partial \mathcal{C}}{\partial \sigma_n} + \frac{\sigma_n(\sigma_{prior}^2 - \sigma_n^2)}{\sigma_{res}^2}$ be the update rule over σ. If the loss curvature can be expressed as $\frac{1}{\sigma_L^2}$, $\sigma_0 < \sigma_L$ and, $\frac{\sigma_{prior}^2}{\sigma_{res}^2} << 1$, then the diagonal elements of the Hessian matrix with respect to μ and σ are given by:*

$$H_D(\mu) = H_D(\sigma) = \lim_{n \to +\infty} \left[\frac{\sigma_{prior}^2}{\sigma_{res}^2 \sigma_n^2} - \frac{1}{\sigma_{res}^2} \right]. \tag{44}$$

Proof. First, let us define: $\alpha_{lim} = \frac{\sigma_{prior}}{\sigma_{res}}$, and $\alpha_n = \sigma_n \sqrt{\frac{1}{\sigma_l^2} + \frac{1}{\sigma_{res}^2}}$ and let us consider the sequence (α_n), defined as:

$$\alpha_{n+1} = \frac{\alpha_n(1 - \alpha_n^2)}{1 - \alpha_{lim}^2}. \tag{45}$$

We will now use a numerical approximation to simplify this sequence. When $\alpha_{lim}^2 << 1$, we have:

$$\alpha_{n+1} = \alpha_n(1 - \alpha_n^2)(1 + \alpha_{lim}^2) \tag{46}$$

$$\sigma_{n+1} = \sigma_n(1 - \frac{\sigma_n^2}{\sigma_L^2} - \frac{\sigma_n^2}{\sigma_{res}^2})(1 + \frac{\sigma_{prior}^2}{\sigma_{res}^2}) \tag{47}$$

$$\sigma_{n+1} = \sigma_n - \frac{\sigma_n^3}{\sigma_L^2} - \frac{\sigma_n^3}{\sigma_{res}^2} + \frac{\sigma_n \sigma_{prior}^2}{\sigma_{res}^2} - \frac{\sigma_n^3 \sigma_{prior}^2}{\sigma_{res}^2 \sigma_l^2} - \frac{\sigma_n^3 \sigma_{prior}^2}{\sigma_{res}^4} \tag{48}$$

$$\sigma_{n+1} = \sigma_n(1 - \frac{\sigma_n^2}{\sigma_L^2}) + \frac{\sigma_n(\sigma_{prior}^2 - \sigma_n^2)}{\sigma_{res}^2} - \frac{\sigma_n^3 \sigma_{prior}^2}{\sigma_{res}^2 \sigma_L^2} - \frac{\sigma_n^3 \sigma_{prior}^2}{\sigma_{res}^4} \tag{49}$$

$$\sigma_{n+1} = \sigma_n(1 - \frac{\sigma_n^2}{\sigma_L^2}(1 + \frac{\sigma_{prior}^2}{\sigma_{res}^2})) + \frac{\sigma_n(\sigma_{prior}^2 - \sigma_n^2(1 + \frac{\sigma_{prior}^2}{\sigma_{res}^2}))}{\sigma_{res}^2}. \tag{50}$$

Again, since $\alpha_{lim}^2 << 1$, this can be further simplified as:

$$\sigma_{n+1} = \sigma_n(1 - \frac{\sigma_n^2}{\sigma_L^2}) + \frac{\sigma_n(\sigma_{prior}^2 - \sigma_n^2)}{\sigma_{res}^2}. \tag{51}$$

We recognize the sequence we obtained in Eq. (6), which was derived from our update rule, the assumption of locally constant curvature and by invoking Lemma 1.

In Appendix B, Proposition 1, we demonstrate that:

$$\lim_{n \to +\infty} \alpha_n = \alpha_{lim}. \tag{52}$$

By squaring the result and reconfiguring the expression, we obtain:

$$\boxed{H_D(\mu) = H_D(\sigma) = \lim_{n \to +\infty} \left[\frac{\sigma_{prior}^2}{\sigma_{res}^2 \sigma_n^2} - \frac{1}{\sigma_{res}^2} \right]}. \tag{53}$$

□

Theorem 2 (Hessian Diagonal via Synaptic Uncertainty Asymptote in Bayesian Neural Networks)**.** *Let $q(\omega|\theta)$ represent the mean-field Gaussian describing the Bayesian neural network, where $\theta = (\mu, \sigma)$ and weight samples are defined as $\omega = \mu + \epsilon \cdot \sigma$, with $\epsilon \sim \mathcal{N}(0, 1)$. Consider $\mathcal{L}$ as the negative log-likelihood and $\mathcal{C}$ as the expectation of $\mathcal{L}$, defined as $\mathcal{C} = \mathbb{E}_{q(\omega|\theta)}[\mathcal{L}(\omega)]$. Let $\sigma_{n+1} = \sigma_n - \sigma_n^2 \times \frac{\partial \mathcal{C}}{\partial \sigma_n}$ be the update rule over σ. If the loss curvature can be expressed as $\frac{1}{\sigma_L^2}$ and $\sigma_0 < \sigma_L$, then the diagonal elements of the Hessian matrix with respect to μ and σ are given by:*

$$H_D(\mu) = H_D(\sigma) = \lim_{n \to +\infty} \frac{1}{2n\sigma_n^2}. \tag{54}$$

Proof. Thanks to the Lemma 1 and the locally constant curvature hypothesis, the update rule over σ becomes: $\sigma_{n+1} = \sigma_n(1 - \frac{\sigma_n^2}{\sigma_L^2})$. By introducing $\alpha_n = \frac{\sigma_n}{\sigma_L}$, we establish the following recurrent sequence:

$$\alpha_{n+1} = \alpha_n(1 - \alpha_n^2) \qquad\qquad 0 < \alpha_0 < 1 \tag{55}$$

In Appendix B, Proposition 2, we demonstrate:

$$\lim_{n \to +\infty} \alpha_n \sqrt{2n} = 1, \tag{56}$$

using $\alpha_n = \frac{\sigma_n}{\sigma_L}$ and $H_D(\mu) = \frac{1}{\sigma_L^2}$. Taking the square of the result we obtain:

$$\boxed{H_D(\mu) = H_D(\sigma) = \lim_{n \to +\infty} \frac{1}{2n\sigma_n^2}}. \tag{57}$$

□

D: Experimental Implementation

Verification of theoretical result The model was trained on MNIST where the input images were scaled between 0 and 1. We employed a fully connected neural network with one hidden layer of 100 neurons, using ReLU activation function. The model was trained over 100,000 mini-batches, where the batch size was 100. The mean values of the Bayesian Neural Network parameter distributions were initialized to the weights obtained from training an equivalent deterministic model - aligning with the hypothesis of locally constant curvature even at $n = 0$. We used the following hyperparameters: $\sigma_0 = 0.1$, $\sigma_{prior} = 0.1$, and $\sigma_{res} = 10$.

Comparison with the Fisher matrix and SI approximation of the Hessian The model was trained on MNIST task in the same fashion as for the verification of the theoretical results, besides the fact the mean values of the Bayesian neural network and the weights of the deterministic neural network were initialized via Kaiming uniform initialization. We used the following hyperparameters, $\sigma_0 = 0.06$, $\sigma_{prior} = 0.06$, and $\sigma_{res} = 10$. For the deterministic neural network, we used a stochastic gradient descent optimizer with lr=0.01.

Benchmark between MESU and BGD The MESU and BGD models were trained on permuted MNIST and the input images were padded to a dimension of 32x32 and then scaled between 0 and 1. We used a fully connected neural network with two hidden layers of 200 neurons, and used ReLU activation function. The batch size was 128. The mean values of the Bayesian neural networks were initialized using Kaiming uniform initialization. We used the following hyperparameters, $\sigma_0 = 0.04$, $\sigma_{prior} = 0.04$, and $\sigma_{res} = 10$.