Uncertainty-Guided Continual Learning in Bayesian Neural Networks – Extended Abstract

Sayna Ebrahimi
UC Berkeley
sayna@eecs.berkeley.edu

Mohamed Elhoseiny
Facebook AI Research
mohamed.elhoseiny@gmail.com

Trevor Darrell
UC Berkeley
trevor@eecs.berkeley.edu

Marcus Rohrbach
Facebook AI Research
maroffm@gmail.com

Abstract

*Continual learning aims to learn new tasks without forgetting previously learned ones. This is especially challenging when one cannot access data from previous tasks and when the model has a fixed capacity. Current regularization-based continual learning algorithms need an external representation and extra computation to measure the parameters' importance. In contrast, we propose Bayesian Continual Learning (*BCL*), where the learning rate adapts according to the uncertainty defined in the probability distribution of the weights in networks. We evaluate our* BCL *approach extensively on diverse object classification datasets with short and long sequences of tasks and report superior or on-par performance compared to existing approaches. Additionally we show that our model can be task-independent at test time, i.e. it does not presume knowledge of which task a sample belongs to.*

1. Introduction

Humans can easily accumulate and maintain can knowledge gained from previously observed tasks, and continuously learn to solve new problems or tasks. Artificial learning systems typically forget prior tasks when they cannot access all training data at once but are presented with task data in sequence. Overcoming these challenges is the focus of *continual learning*.

Catastrophic forgetting [12] refers to the significant drop in the performance of a learner when switching from a trained task to a new one. This phenomenon occurs because trained parameters on the initial task change in favor of learning new objectives. Given a network of limited capacity, one way to address this problem is to identify the importance of each parameter and penalize further changes to those parameters that were deemed to be important for the previous tasks [6, 1, 19].

Bayesian neural networks [2] propose an intrinsic importance model based on weight uncertainty. These networks represent each parameter with a distribution defined by a mean and variance over possible values drawn from a shared latent probability distribution. Variational inference can approximate posterior distributions using Monte Carlo sampling for gradient estimation. These networks act like ensemble methods in that they reduce the prediction variance but only use twice the number of parameters present in a regular neural network

We propose Bayesian Networks for continual learning, and develop a new method which exploits the inherent measure of uncertainty therein to adapt the learning rate of individual parameters. Second, we present a hard-threshold variant of our method that decides which parameters to freeze. Third, we validate our approach experimentally, comparing it to prior art both on single datasets split into different tasks, as well as for the more difficult scenario of learning a sequence of different datasets. Forth, in contrast to most prior work, our approach does not rely on knowledge about task boundaries at inference time, which humans do not need and might not be always available. We show that our approach naturally supports this scenario, sometimes also referred to as a "single head" scenario for all tasks. We refer to evaluation metric of a "single head" model without task information at test time as "generalized accuracy".

1.1. Background: Variational Bayes-by-Backprop

Let $\mathbf{x} \in \mathbb{R}^n$ be a set of observed variables and $\mathbf{w}$ be a set of latent variables. A neural network, as a probabilistic model $p(\mathbf{y}|\mathbf{x}, \mathbf{w})$, given a set of training examples $\mathcal{D} = (\mathbf{x}, \mathbf{y})$ can output $\mathbf{y}$ which belongs to a set of classes by using the set of weight parameters $\mathbf{w}$. We first assume

a family of probability densities over the latent variables $\mathbf{w}$ parametrized by θ, i.e., $q(\mathbf{w}|\theta)$. We then find the closest member of this family to the true conditional probability of interest $p(\mathbf{w}|\mathbf{x},\mathbf{y})$ by minimizing the Kullback-Leibler (KL) divergence between q and p:

$$q^*(\mathbf{w}|\theta) = \arg\min_\theta \mathrm{KL}\big(q(\mathbf{w}|\theta)\|p(\mathbf{w}|\mathbf{x})\big) \quad (1)$$

Eq. 1 is commonly known as variational free energy or expected lower bound:

$$\mathcal{L}(\theta,\mathcal{D}) = \mathrm{KL}\big[q(\mathbf{w}|\theta)\|p(\mathbf{w})\big] - \mathbb{E}_{q(\mathbf{w}|\theta)}\big[\log(p(\mathbf{y}|\mathbf{x},\mathbf{w}))\big] \quad (2)$$

[2] showed that Eq. 2 can be approximated using N Monte Carlo samples from the variational posterior:

$$\mathcal{L}(\theta,\mathcal{D}) \approx \sum_{i=1}^{N} \log q(\mathbf{w}^{(i)}|\theta) - \log p(\mathbf{w}^{(i)}) - \log(p(\mathbf{y}|\mathbf{x},\mathbf{w}^{(i)})) \quad (3)$$

We assume $q(\mathbf{w}|\theta)$ to have a Gaussian pdf with diagonal covariance and parametrized by $\theta = (\mu,\rho)$. A sample weight of the variational posterior can be obtained by sampling from a unit Gaussian and reparametrized by $\mu + \sigma \circ \epsilon$ where $\circ$ is a pointwise multipliation. Standard deviation is parametrized as $\sigma = \log(1+\exp(\rho))$ and thus is always non-negative.

2. Bayesian Continual Learning Neural Networks

A common strategy to perform continual learning is to reduce forgetting by regularizing further changes in the model representation based on parameters' *importance*. In this section we introduce our Bayesian Continual Learning approach (BCL), which exploits estimated uncertainty of the parameters' posterior distribution to regulate the change in certain/uncertain parameters. BCL regulates the change of certain/important parameters in a soft way such that the learning rate of each parameter and hence its gradient update becomes a function of its *importance*.

Uncertainty-defined *importance* We use the well-defined uncertainty in parameters distribution, i.e., standard deviation, as a notion of *importance*. In particular we scale the learning rate for each parameter proportional to its uncertainty to reduce changes in certain parameters while allowing uncertain parameters to alter in favor of learning new tasks.

Parameter regularization in BCL. Learning rate regularization means that the learning rate is adapted per parameter according to an importance measure Ω as shown in the following equations

$$\mu' = \mu - \Omega_\mu \alpha \nabla \mathcal{L}_{BBB_\mu} \quad (4)$$
$$\rho' = \rho - \Omega_\rho \alpha \nabla \mathcal{L}_{BBB_\rho} \quad (5)$$

where α is the learning rate and $\nabla\mathcal{L}_{BBB_\mu}$ and $\nabla\mathcal{L}_{BBB_\rho}$ are the gradients of Eq. 3 w.r.t μ and ρ, respectively. We empirically find that $\Omega_\sigma = \sigma$ and $\Omega_\rho = 1$, i.e. no change in the learning for ρ works best.

2.1. BCL using weight pruning (BCL-P)

A variant of our method, BCL-P, is related to recent efforts in weight pruning in the context of reducing inference computation and network compression [9]. Forgetting is prevented in pruning by saving a task-specific binary mask of important vs. unimportant parameters. Here, we adapt pruning to the Bayesian neural networks and propose to use the statistically-grounded uncertainty defined in Bayesian neural networks as the pruning criterion. We use the signal-to-noise ratio (SNR) [2] for each parameter defined as $\mathrm{SNR} = \frac{|\mu|}{\sigma}$ as a hard threshold for importance.

3. Experimental Setup and Results

Datasets and sequence of tasks: We evaluate our approach in two common scenarios for continual learning: 1) class-incremental learning of a single or two randomly alternating datasets, where each task covers only a subset of the classes in a dataset, and 2) continual learning of multiple datasets, where each task is a dataset. We use MNIST split, permuted MNIST, and CIFAR10/100 for class incremental learning with similar experimental settings as used in [16, 10]. We also evaluate our approach on continually learning a sequence of datasets which have different distributions using the identical 8 task sequence as in [16], which includes FaceScrub [13], MNIST, CIFAR100, NotMNIST [3], SVHN, CIFAR10, TrafficSigns [17], and FashionMNIST [18].

Baselines: Within the Bayesian framework, we compare to three models which do not incorporate the importance of parameters, namely fine-tuning, feature extraction, and joint training. In fine-tuning (BBB-FT), training continues upon arrival of new tasks without any forgetting avoidance strategy. Feature extraction, denoted as (BBB-FE) in our experiments, refers to freezing all layers in the network after training the first task and training only the last layer for the remaining tasks. In joint training (BBB-JT) we learn all the tasks jointly in a multitask learning fashion which serves as the upper bound for average accuracy on all tasks, as it does not adhere to the continual learning scenario. From prior work, we compare with state-of-the-art approaches including Elastic Weight Consolidation (EWC) [6], Incremental Moment Matching (IMM) [7], Learning Without Forgetting (LWF) [8], Less-Forgetting Learning (LFL) [5], PathNet [4], Progressive neural networks (PNNs) [15], and Hard Attention Mask (HAT) [16] using implementations provided by [16].

Performance measurement: Let n be the total number of tasks. Once all are learned, we evaluate our model on all

Table 1: Continually learning on different datasets. BWT and ACC in %. (*) denotes that methods do not adhere to the continual learning setup: BBB-JT and ORD-JT serve as the upper bound for ACC for BBB/ORD networks, respectively. ‡ denotes results reported by [16]. † denotes the result reported from original work. BWT was not reported in ‡ and †. All others results are (re)produced by us.

(a) Split MNIST, in two tasks.

Method	BWT	ACC
PackNet [11]	0.0	98.9
LWF [8]	−0.2	99.1
HAT [16]	0.0	99.0
ORD-FT	−6.8	92.4
ORD-FE	0.0	97.9
BBB-FT	−0.6	98.4
BBB-FE	0.0	98.0
BCL-PRN (Ours)	0.0	99.0
BCL-LR (Ours)	0.0	**99.2**
ORD-JT*	0.0	99.1
BBB-JT*	0.0	99.5

(b) Permuted MNIST, 10 random permutations.

Method	#Params	BWT	ACC
GEM [10]‡	0.1M	-	82.6
SI [19]‡	0.1M	-	86.0
EWC [6]‡	0.1M	-	88.2
VCL [14]†	0.1M	-	90.0
HAT [16]‡	0.1M	-	91.6
BCL-LR (Ours)	0.1M	−0.4	91.4
LWF [8]	1.9M	−31.2	65.7
IMM [7]	1.9M	−7.1	90.5
HAT [16]	1.9M	0.0	97.3
BBB-FT	1.9M	−0.6	90.0
BBB-FE	1.9M	0.0	93.5
BCL-PRN (Ours)	1.9M	−0.9	97.2
BCL-LR (Ours)	1.9M	0.0	**97.4**
BBB-JT*	1.9M	0.0	98.1

(c) Alternating Incremental CIFAR10/100

Method	BWT	ACC
PathNet [4]	0.0	28.9
LWF [8]	−37.9	42.9
LFL [5]	−24.2	47.7
IMM [7]	−12.2	69.4
PNN [15]	0.0	70.7
EWC [6]	−1.5	72.5
HAT [16]	0.0	78.3
BBB-FE	0.0	51.0
BBB-FT	−7.4	68.9
BCL-PRN (Ours)	−1.9	77.3
BCL-LR (Ours)	−0.7	**79.4**
BBB-JT*	1.5	83.9

n tasks. ACC is the average test classification accuracy across all tasks. To measure forgetting we report backward transfer, BWT, which indicates how much learning new tasks has influenced the performance on previous tasks. While BWT < 0 directly reports *catastrophic forgetting*, BWT > 0 indicates that learning new tasks has helped with the preceding tasks. Formally, BWT and ACC are defined as follows:

$$\text{BWT} = \frac{1}{n}\sum_{i=1}^{n} R_{i,n} - R_{i,i}, \quad \text{ACC} = \frac{1}{n}\sum_{i=1}^{n} R_{i,n} \qquad (6)$$

where $R_{i,n}$ is the test classification accuracy on task i after sequentially finishing learning the n^{th} task. Note that in BCL-P, $R_{i,i}$ refers the test accuracy on task i before pruning and $R_{i,n}$ after pruning which is equivalent to the end of sequence performance.

Table 1 and 2 show ACC and BWT of BCL and BCL-P in comparison to state-of-the-art models. BCL performs better than all baselines reaching ACC of 97.7%, 97.4%, 70.4%, and 84.0% in Split MNIST, permuted MNIST, CIFFAR10/100, and 8 subsequent tasks, respectively.

Single Head and Generalized Accuracy. BCL can be used even if the task information is not given at test time. For this purpose, at training time, instead of using a separate fully connected classification head for each task, we use a single head with total number of outputs for all tasks.

Table 3 presents our results for BCL and BBB-FT trained with a single head against having a multi-head architecture, in columns 4-7. Interestingly, we found a small performance degrade for BCL from training with multi head to single head. We evaluated BCL and BBB-FT with a more challenging metric where the prediction space covers the classes across all the tasks. Hence, confusion of of similar class labels across tasks can be measured. Performance for this condition is reported as Generalized ACC in Table 3 in columns 2-3. We also observed a small performance reduction in going from ACC to Generalizd ACC, suggesting non-significant confusion caused by presence of higher number of classes at test time.

Table 2: Sequence of 8 tasks.

Method	BWT (%)	ACC (%)
LFL [5]	−10.0	8.6
PathNet [4]	0.0	20.2
LWF [8]	−54.3	28.2
IMM [7]	−38.5	43.9
EWC [6]	−18.0	50.7
PNN [15]	0.0	76.8
HAT [16]	−0.1	81.6
BBB-FT	−23.1	43.1
BBB-FE	0.0	58.1
BCL-PRN (Ours)	−2.5	80.4
BCL-LR (Ours)	−0.8	**84.0**
BBB-JT*	−1.2	84.1

Table 3: Single Head vs. Multi Head architecture and Generalized vs. Standard Accuracy. Generalized accuracy means that Test time task independent task information is not available at test time. SM, PM, CF, and 8T denote the Split MNIST, Permuted MNIST, Alternating CIFAR10/100, and sequence of 8 tasks, respectively.

	Generalized ACC		ACC			
	Single Head		Single Head		Multi Head	
Exp	BCL-LR	BBB-FT	BCL-LR	BBB-FT	BCL-LR	BBB-FT
SM	98.7	98.1	98.9	98.7	99.2	98.4
PM	92.5	86.1	95.1	88.3	97.7	90.0
CF	71.2	65.2	74.3	67.8	79.4	68.9
8T	76.8	47.6	79.9	53.2	84.0	43.1

4. Conclusion

In this work, we proposed a continual learning formulation based on Bayesian neural networks, called BCL, that uses uncertainty predictions to perform continual learning: important parameters can be either fully preserved through a saved binary mask (BCL-P) or allowed to change condi-

tioned on their uncertainty for learning new tasks (BCL). We show that BCL performs superior or on par with state-of-the-art models such as HAT [16] across all the experiments. BCL can also be used in a single head setting where the right subset of classes belonging to the task is not known during inference.

References

[1] 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. 1

[2] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1613–1622. PMLR, 2015. 1, 2

[3] Yaroslav Bulatov. Notmnist dataset. *Google (Books/OCR), Tech. Rep.[Online]. Available: http://yaroslavvb. blogspot. it/2011/09/notmnist-dataset. html*, 2011. 2

[4] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017. 2, 3

[5] Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. Less-forgetting learning in deep neural networks. *arXiv preprint arXiv:1607.00122*, 2016. 2, 3

[6] 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*, page 201611835, 2017. 1, 2, 3

[7] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*, pages 4652–4662, 2017. 2, 3

[8] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, pages 614–629. Springer, 2016. 2, 3

[9] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2755–2763. IEEE, 2017. 2

[10] David Lopez-Paz et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017. 2, 3

[11] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[12] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 1

[13] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 343–347. IEEE, 2014. 2

[14] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. In *ICLR*, 2018. 3

[15] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 2, 3

[16] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4548–4557. PMLR, 2018. 2, 3, 4

[17] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1453–1460. IEEE, 2011. 2

[18] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 2

[19] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995. PMLR, 2017. 1, 3