A Bayesian Approach to Incremental Learning

Yanwei Fu [1] **Xianglong Hu** [2]

Abstract

Most research nowadays focuses mainly on closed problems in which all the classes are given. However, in reality, human intelligence constantly learn new concepts based on the old ones. As incremental learning is getting more attention these days, we propose a new idea to approach this problem, adapting Bayesian variational inference to incremental learning. Unlike traditional neural networks, variational inference could capture the correlations of the data. We try to show in this paper that this approach is superior in the following ways. First, it prevents catastrophic forgetting. Second, the memory grows linearly in the class number. Third, the old knowledge facilitates the learning of new classes.

1. Introduction

The inherent structure of knowledge is actually incremental, which coincides the way how we learn it. To be specific, human intelligence first builds up what we called common sense and then understands the world stretched from that base. However, little attention has been paid to this field. As the field of computer vision is getting closer to artificial intelligence, it is apparent that more flexible techniques are required to handle the large-scale and dynamic properties of real-world object categorization problems. (Rudd et al., 2017) has addressed these problems as *open set problems*, while (Rebuffi et al., 2017) further refines this process with the concept *class-incremental learning*.

As explained in 1 and 2, the crucial part toward open world recognition is to incorporate new classes whenever the training date is ready. Intuitively, one could always periodically retrain classifiers. However, this is computationally expensive. Online stream classifiers may be adapted. Nevertheless, this always leads to the quick deterioration of classification of accuracy, often known as *catastrophic forgetting* or *catastrophic interference*.

To better solve these problems, (Rebuffi et al., 2017) formalizes three rules for an algorithm to be qualified as class-incremental:

1. it should be trainable from a stream of data in which examples of different classes occur at different times,
2. it should at any time provide a competitive multi-class classifier for the classes observed so far,
3. its computational requirements and memory footprint should remain bounded, or at least grow very slowly, with respect to the number of classes seen so far.

It also introduces a practice strategy in the class-incremental setting. Classification is based on *nearest-mean-of-exemplars*, where exemplars are selected from each class. By storing the exemplars, the nearest neighbour method bypasses the trap of catastrophic forgetting. This method is not satisfactory enough in that it provides little characterization of the statistical distribution of the data. Moreover, the nearest neighbour method assumes the data is unimodally distributed, which greatly limits its application. In comparison, our work not only gives a good characterization of the data distribution, but also adapts well to multi-modally distributed data in addition to the satisfaction of the three criteria mentioned above. Finally, since Bayesian statistics is used in our model, predictions could be made with uncertainty, which reflects our confidence level. Prediction with confidence could further help to recognize new classes. Whenever the uncertainty is high enough to be over some threshold, this certain instance could be labeled as unknown.

2. Related Work

Weight Uncertainty in Neural Networks: Plain feed-forward neural networks are prone to overfitting. Thus the Bayesian inference was introduced. In general, exact Bayesian inference on the weights of a neural network is

[1]School of Data Science, Fudan University, Shanghai, China [2]Courant Institute of Mathematical Science, New York University, New York, USA. Correspondence to: Yanwei Fu <yanweifu@fudan.edu.cn>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

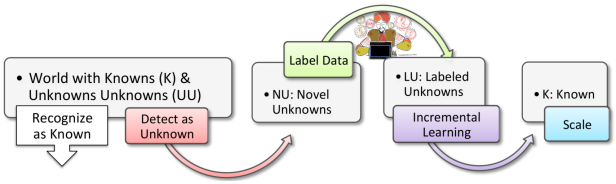

Figure 1. In open world recognition, the system must be able to recognize objects and associate them with known classes while also being able to label classes as unknown. These novel unknowns must then be collected and labeled (e.g. by humans). When there are sufficient labeled unknowns for new class learning, the system must incrementally learn and extend the multi-class classifier, thereby making each new class known to the system. Open World recognition moves beyond just being robust to unknown classes and toward a scalable system that is adapting itself and learning in an open world.

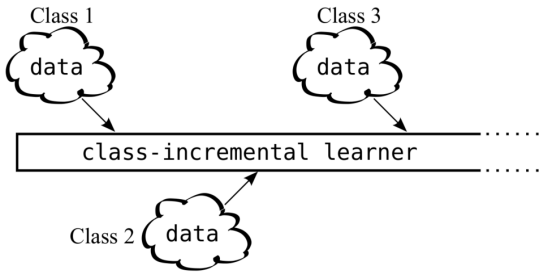

Figure 2. Class-incremental learning: an algorithm learns continuously from a sequential data stream in which new classes occur. At any time, the learner is able to perform multi-class classification for all classes observed so far.

intractable as the number of parameters is very large and the functional form of a neural network does not lend itself to exact integration. (Graves, 2011) first gives a practical Monte Carlo method to variational inference with back propagation. Based on that, (Blundell et al., 2015) further extend the priors to non-Gaussian priors. At the stage of prediction and back propagation, (Kingma et al., 2015) further improves the efficiency of sampling by changing the estimator. This trick is named *local reparameterization*. Our work follows their techniques.

Hierarchical Bayesian Neural Network: Hierarchical Bayesian domain adaptation has been proposed to improve the performance of multiple domains of the same task (Finkel & Manning, 2009). (Joshi et al., 2017) extended this line of work by putting forward Bayesian neural networks to solve gesture recognition problems by assigning different subjects independent variances. In 3, different W_g shares the same W_0, the mean, but with subject-variant variances. W_0 keeps the correlations of various networks while subject-variant differences are captured. As in 4, the weights of a new network is learned by a few instances of training data from a new subject. We found this network structure suitable to class-incremental learning, especially when the classes are correlated.

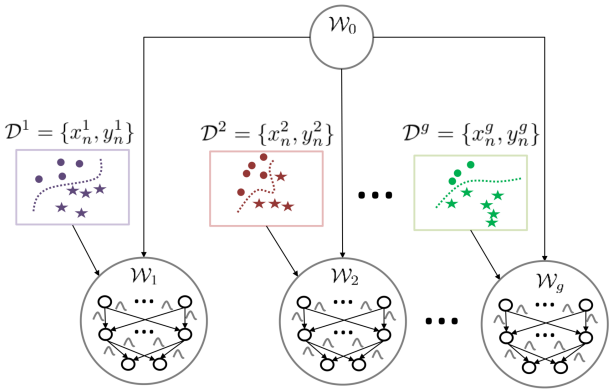

Figure 3. Given gesture examples produced by g subjects, we train a classifier using a hierarchical framework, where W_g is the set of group-specific weights parameterizing a Bayesian neural network. The different shapes correspond to different gesture classes and the different colors represent the subjects who produced those examples

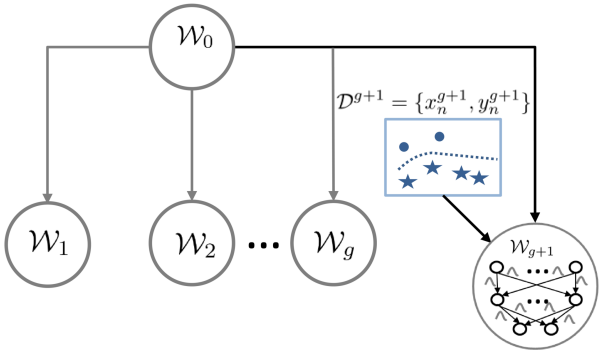

Figure 4. Given few instances of training data from a new subject, we personalize our model to learn weights specific to the new subject.

3. Method

This part is still in draft, and its majority would just be the transportation from (Joshi et al., 2017).

4. Experiment

In this section, we briefly describe our current results and difficulties, as well as comparisons with *iCaRL* (Rebuffi et al., 2017). The benchmark standard is inherited from it too.

Benchmark protocol: Currently the classes are arranged in a fixed random order. [1] Each method is then trained in a class-incremental way on the available training data. After each batch of classes, the resulting classifier is evaluated on the test part data of the dataset, *considering only those classes that have already been trained* .

Datasets: We use CIFAR-10 and CIFAR-100. For CIFAR-10, we train all 10 classes only in a batch of one class[2]. Since each network only categorizes one class, the training data are composed of two parts, positive and negative samples. The positive samples are just the training data of that class, yet the negative ones are randomly sampled from all the other classes and labeled to be different from the positive ones. For CIFAR-100, all 100 classes are trained in batches of 10, 20 and 50 at a time. The training data are unprocessed. We compare our results with (Rebuffi et al., 2017) on CIFAR-100 only, since (Rebuffi et al., 2017) doesn't experiment on CIFAR-10.

Implementation: We rely on pytorch and extract the features of size 456 from a 40-layers Densenet. Our hierarchical Bayesian neural network consists of two fully connected layers. The loss function is the same as the one in (Joshi et al., 2017) except that the terms related to the data are amplified due to dataset differences. The learning rate changes as follows, $a \cdot (b + (epoch + 1))^{decay}$. We are still exploring the best parameters and learning rates so the final parameters are not settled.

4.1. Results

Both 5 and 6 trains on CIFAR-10. 6 differs from 5 in the loss function. 6 trains like ordinary Bayesian neural networks, all the other terms in the loss functions are set to zero except the ones that are directly related to the data. The motivation for this comparison experiment is to examine how the hierarchical structure boosts the training. In other words, how W_0 helps capture the variations between different W_g in 4. However, 5 and 6 are almost the same. Probably the network is too small. Or classes within CIFAR-10 are poorly correlated. This requires further examination in CIFAR-100.

[1] However, this might be subject to possible changes for our network should perform better when classes are correlated in theory, which means similar classes should be grouped together. We haven't moved to this stage yet.

[2] This reminds me of doing more experiments on batch 2 and 5 . . .

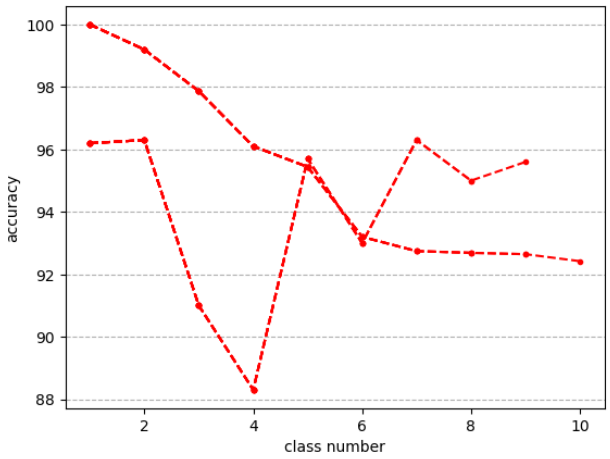

Figure 5. This experiment consists of two layers and trains for 150 epochs each class. In 13, $a = 0.07, b = -0.5, decay = -0.55$. The upper line is overall accuracy for all the known classes while the lower one is accuracy for the current class batch.

7 and 9 trains on CIFAR-100 with a class batch of 10 and 20 respectively. Compared with 11 and 12 from the results of (Rebuffi et al., 2017), our results are superior to all the other methods except *iCaRL*. However, from 8 and 10, it is obvious that we haven't found appropriate parameters and there is still room for improvement. It remains unclear whether it is possible to surpass *iCaRL*. However, I am not positive toward that since there is a 10% gap.

4.2. Difficulties

From 8 10, we are still far from satisfactory results. At the moment, we've tried various learning rates. But the loss is susceptible to divergence and no training at all. The loss would divergence when the learning rate is either small or large. We've been stuck here for a while.

References

Blundell, Charles, Cornebise, Julien, Kavukcuoglu, Koray, and Wierstra, Daan. Weight uncertainty in neural networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1613–1622. JMLR.org, 2015.

Finkel, Jenny Rose and Manning, Christopher D. Hierarchical bayesian domain adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pp. 602–610, Stroudsburg, PA, USA, 2009. Association for

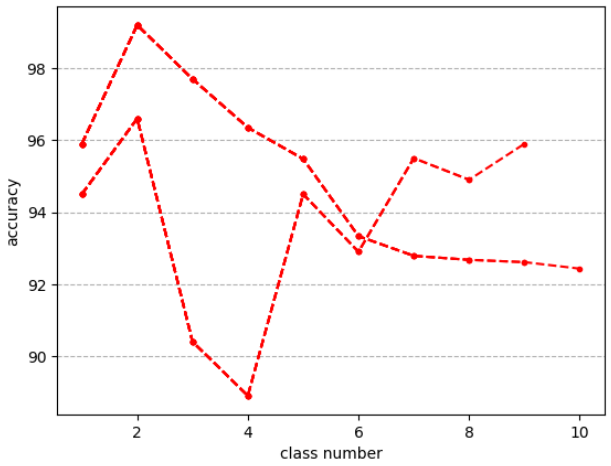

Figure 6. This experiments follows the same parameters as 5 while the loss differs.

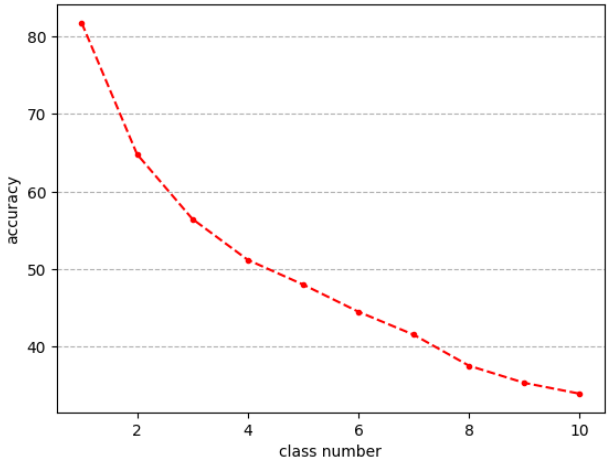

Figure 7. This experiment consists of two layers and trains for 150 epochs each class batch. The class batch is 10. In 13, $a = 0.07, b = -0.5, decay = -0.55$. This is overall accuracy for all the classes. The x coordinate is actually the network number. There are 100 classes and 10 networks.

Computational Linguistics. ISBN 978-1-932432-41-1.

Graves, Alex. Practical variational inference for neural networks. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 2348–2356. Curran Associates, Inc., 2011.

Joshi, A., Ghosh, S., Betke, M., Sclaroff, S., and Pfister, H. Personalizing gesture recognition using hierarchical bayesian neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 455–464, July 2017. doi: 10.1109/CVPR.2017.56.

Kingma, Diederik P, Salimans, Tim, and Welling, Max. Variational dropout and the local reparameterization trick. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 2575–2583. Curran Associates, Inc., 2015.

Rebuffi, Sylvestre-Alvise, Kolesnikov, Alexander, Sperl, Georg, and Lampert, Christoph H. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5533–5542, 2017. doi: 10.1109/CVPR.2017.587.

Rudd, E. M., Jain, L. P., Scheirer, W. J., and Boult, T. E. The extreme value machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2017.2707495.

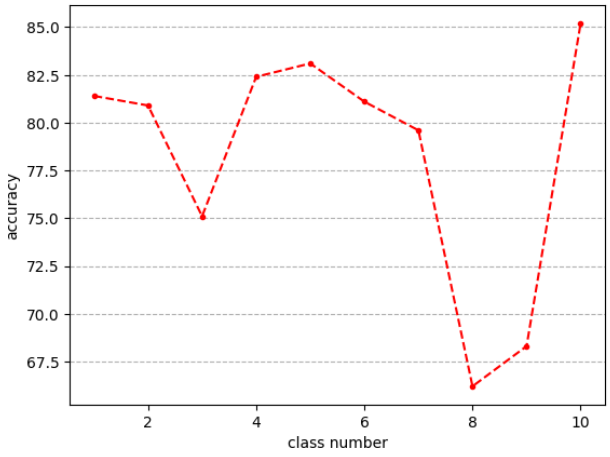

Figure 8. This is the same experiment as 7. Single class batch accuracy is plotted.

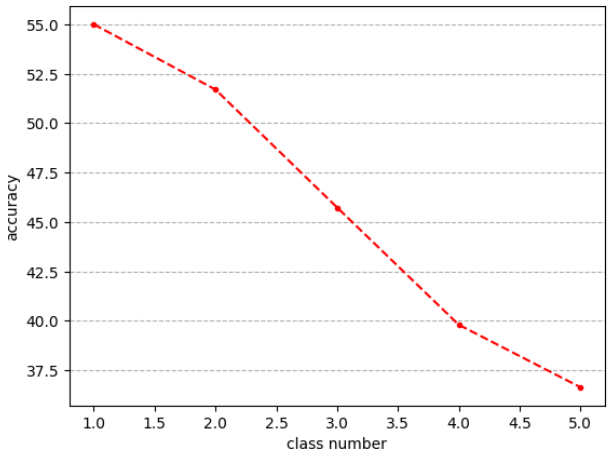

Figure 9. This experiment consists of two layers and trains for 150 epochs each class batch. The class batch is 20. In 13, $a = 0.07, b = -0.5, decay = -0.55$. This is overall accuracy for all the classes. The x coordinate is actually the network number. There are 100 classes and 5 networks.

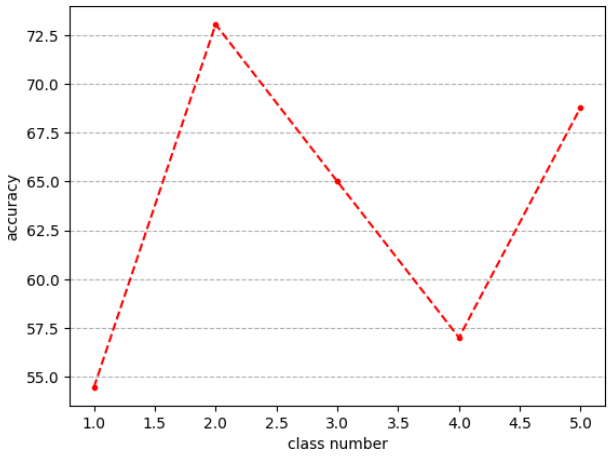

Figure 10. This is the same experiment as 9. Single class batch accuracy is plotted.

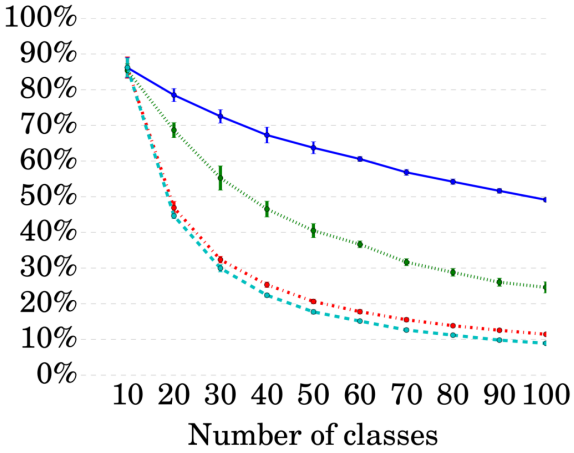

Figure 11. Results from (Rebuffi et al., 2017) with class batch of 10.

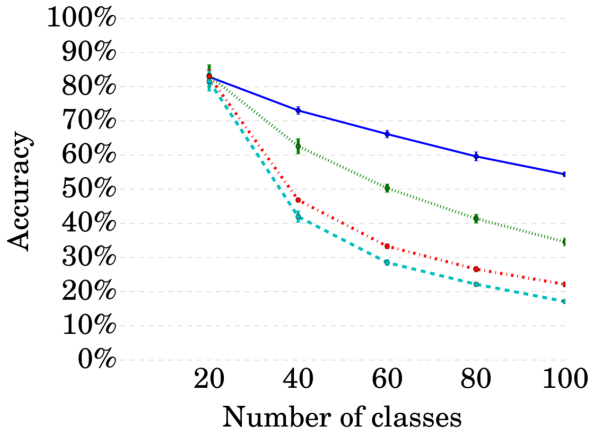

Figure 12. Results from (Rebuffi et al., 2017) with class batch of 20.

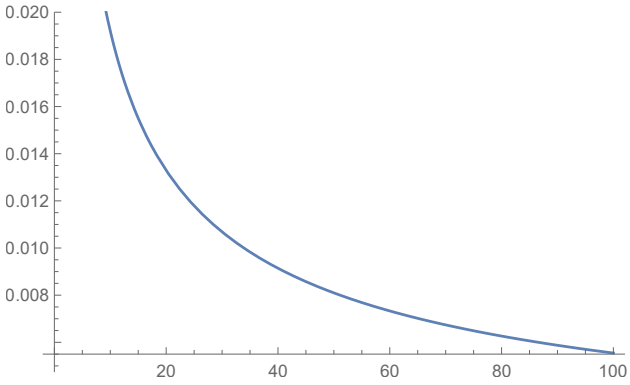

Figure 13. Current working learning rate.