One Pass ImageNet

Huiyi Hu[1], **Ang Li**[1], **Daniele Calandriello**[2], **and Dilan Gorur**[1]

[1]DeepMind, Mountain View, USA
[2]DeepMind, Paris, France

{clarahu,anglili,dcalandriello,dilang}@deepmind.com

Abstract

We present the One Pass ImageNet (OPIN) problem, which aims to study the effectiveness of deep learning in a streaming setting. ImageNet is a widely known benchmark dataset that has helped drive and evaluate recent advancements in deep learning. Typically, deep learning methods are trained on static data that the models have random access to, using multiple passes over the dataset with a random shuffle at each epoch of training. Such data access assumption does not hold in many real-world scenarios where massive amounts of data are collected from a stream and storing and accessing all the data becomes impractical due to storage costs and privacy concerns. For OPIN, we treat the ImageNet data as arriving sequentially, and there is limited memory budget to store a small subset of the data. We observe that training a deep network in a single pass with the same training settings used for multi-epoch training results in a huge drop in prediction accuracy. We show that the performance gap can be significantly decreased by paying a small memory cost and utilizing techniques developed for continual learning, despite the fact that OPIN differs from typical continual problem settings. We propose using OPIN to study resource-efficient deep learning.

1 Introduction

ImageNet [1] is one of the most influential benchmarks used to evaluate improvements in machine learning research. Tremendous progresses have been made over the past decade in terms of a model's accuracy on the ImageNet dataset. While improving model accuracy on ImageNet has been one of the major focuses in the past, much less effort has been dedicated to evaluating if resource efficient supervised learning algorithms can achieve such high accuracy on ImageNet. Most existing supervised learning methods assume the data is i.i.d., static and pre-existing. They train models with multiple epochs, i.e., multiple passes of the whole dataset. Specifically, a top-performing Residual Neural Network [2] is trained with 90 epochs; i.e., the model needs to review each of the 1.2M examples 90 times. One natural question to ask is whether so many passes over the whole data are necessary to train a model.

We are also motivated by the fact that real world data often comes hourly or daily in a stream and in a much larger scale. Maintaining all the data in a storage can be expensive and probably unnecessary. Additionally, real world data sometimes contains private information from human users, which further restricts the possibility of saving all the data into a separate storage. Without a pre-recorded dataset, the popular multi-epoch training method becomes impractical to these real world scenarios.

We propose the One Pass ImageNet (OPIN) problem to study the resource efficiency of deep learning in a streaming setting with constrained data storage, where space complexity is considered an important evaluation metric. The goal is to develop a system that trains a model where each example is passed to the system only once. There is a small memory budget but no restriction on how the system utilizes its memory; it could store and revisit some examples but not all. Unlike the

Table 1: Comparing One-Pass methods with Multi-epoch training in accuracy, storage, compute metrics. Naive One-Pass method results in significant accuracy drop. A priority replay based method achieves better accuracy while maintaining the significant improvements on storage and compute metrics over multi-epoch training. Higher accuracy, lower storage and lower compute are better.

	Accuracy (%) ↑	**Storage (%)** ↓	**Compute (%)** ↓
Multi-epoch (90 epochs)	76.9	100	100
One-Pass (Naive)	30.6	0	1.1
One-Pass (Prioritized Replay)	65.0	10	10

task-incremental continual learning setting [3, 4], the One-Pass ImageNet problem does not have a special ordering of the data, nor is there a specific distribution shift as in task-free continual learning [5–7]. That is, the data comes from a fixed uniform random order. We use ResNet-50 [2] in all our experiments, and leave the question of choice of architecture as future work.

We observe that training a ResNet-50 [2] in a single pass leads to only 30.6% top-1 accuracy on the validation set, a significant drop from 76.9% top-1 accuracy obtained from the common 90-epoch training[1]. Inspired by the effectiveness of memory-based continual learning [4, 8–10], we propose an error-prioritized replay (EPR) method to One Pass ImageNet. The proposed approach utilizes a priority function based on predictive error. Results show that EPR achieves 65.0% top-1 accuracy, improving over naive one-pass training by 34.4%. Although it still performs 11.9% lower than multi-epoch training in terms of accuracy, EPR shows superior resource efficiency which reduces total gradient update steps by 90% and total required data storage by 90%.

We believe OPIN is an important first step that allows us to understand how existing techniques can train models in terms of computation and storage efficiency, although it may not be the most realistic example for the data streaming setting. We hope our results could inspire future research on large scale benchmarks and novel algorithms on resource-efficient supervised learning.

2 One Pass ImageNet

The One Pass ImageNet problem assumes that examples are sent in mini-batches and do not repeat. The training procedure ends when the whole dataset is revealed. No restriction is applied on how the algorithm utilizes its own memory, so a memory buffer that records past examples is allowed. However, the algorithm's data storage size is a major evaluation metric to evaluate space efficiency.

We perform our study on a commonly used ImageNet setup: a ResNet-50 [2] trained over 90 epochs with cosine learning rate and augmented examples. We refer to this method as *Multi-epoch* throughout the paper. The images are preprocessed by resizing to 250×250 and then performing augmentation into a size of 224×224. During training, the augmentation includes random horizontal flipping and random cropping. At test time, only center cropping is applied to the images.

Evaluation metrics. While standard ImageNet benchmark focuses on a model's overall accuracy, the One-Pass ImageNet problem aims at studying the learning capability under constrained space and computation. So the problem becomes essentially a multi-objective problem. We propose to evaluate training methods using three major metrics: (1) *accuracy*, represented by the top-1 accuracy in the test set, (2) *space*, represented by total additional data storage needed, and (3) *compute*, represented by the total number of global steps for back-propagation. The space and compute metric is calculated relative to the multi-epoch training method, i.e., both metrics for Multi-epoch method are 100%. The Multi-epoch method needs to save all the data into storage, so the space metric is measured by the size of the data storage divided by the size of the dataset. The Multi-epoch method needs to train a model with 90 epochs (or 100M global steps), so the compute metric is measured by the total number of back-propagation operations divided by 100M.

Naive baseline. A simple baseline method for the One Pass problem is to train a model with the same training configuration that multi-epoch training uses but with only a single epoch, which we call *Naive One-Pass*. Since each example is seen only once, we replace the random augmentation with center cropping (which is used in the model evaluation) in this Naive baseline. Table 1 shows a comparison

[1] The top-1 accuracy is obtained with 1-crop evaluation in ImageNet.

between multi-epoch training and naive one-pass training measured in three metrics: accuracy, space and compute. All metrics are in percentage. While Naive One-Pass is significantly worse than multi-epoch in terms of accuracy, its space and compute efficiency are both significantly higher. The Naive One-Pass does not need to save any data examples into memory and at the same time it only trains for one-epoch, the total number of training steps is $1/90 \approx 1.1\%$ that of multi-epoch training.

Problem Characteristics. Here we list four properties of the OPIN problem as below:

1. *The cold-start problem*: Model start from random initialization. So the representation learning becomes challenging in OPIN especially during the early stage of the training.
2. *The forgetting problem*: Each example is passed to the model only once. Even though the data is i.i.d., vanilla supervised learning is likely to incur forgetting of early examples.
3. *A natural ordering of data*: No artificial order of the data is enforced. So the data can be seen as i.i.d., which is different from many existing continual learning benchmarks.
4. *Multiple objectives*: The methods are evaluated using three metrics (accuracy, space and compute), so the goal is to improve all three metrics in a single training method.

3 A Prioritized Replay Baseline

Memory replay is a common approach in continual learning [9]. Existing works have shown that memory replay is effective in sequential learning [11, 4]. As our first investigation of the One Pass ImageNet problem, we study how a replay buffer could improve the overall performance in OPIN.

3.1 Replay buffer

Replay buffer is an extra memory that explicitly saves the data. This memory usually has a very limited size. At each training step, the received mini-batch of examples is inserted into this replay memory. Since the buffer size is smaller than the whole dataset, the typical solution is to apply a reservoir sampling strategy [12] where each example is inserted at a probability of $p(n, m) = m/n$, where m is the memory size and n the total number of seen examples. In order to promote inclusion of fresh examples[2], we multiply the inclusion probability by a β factor, i.e., $p(n, m) = \beta m/n$ (we choose $\beta = 1.5$), so that more recent examples will be more likely to be included in the memory.

At each training step, the model is trained on the incoming mini-batch combined with extra examples sampled from the replay buffer. Existing research on continual learning has shown that uniform sampling from a replay buffer is effective in many cases [4]. However, a common intuition is that examples are not equally important when being replayed. The idea of prioritized experience replay [15] is to add a priority score to each example and sample from the buffer according to the probability distribution normalized from the priority scores.

In order to apply data augmentation to replay examples, we save jpeg-encoded image bytes into the replay buffer instead of image tensors, which turns out to be more space efficient (3x more examples can be saved under the same memory budget). We found replaying multiple examples at each step could dramatically improve the model accuracy, although with a trade-off in compute. For each step when we receive one mini-batch of images, we replay k mini-batches from the replay buffer, which leads to $k + 1$ epochs of compute effectively.

3.2 Priority function

We study a prioritized replay method that uses the predictive error (loss value) as the priority function, which we call Error-Prioritized Replay (EPR). The priority function is defined as

$$P(x, y) = 1 - \alpha e^{-\ell(x,y;\theta)} \tag{1}$$

where $\ell(x, y; \theta)$ is the loss value given input example x, ground truth label y and model parameter θ. The smoothing factor α varies from 0 to 1 such that $\alpha(T) = 1 - cos(T/T_{\text{max}})$ where T is the current global step and T_{max} is the maximum global step. We choose a smoothed α because the model's prediction is not trustworthy at the early stage of training. When the loss function is the

[2] Similar ideas on encouraging recent examples in reservoir sampling can be found in [13, 14].

Table 2: One-Pass ImageNet evaluation: Computation, Memory and Accuracy. The effective epoch is computed as the number of replay steps added by 1. The compute metric is calculated by the effective epoch divided by 90 (compared against the 90-epoch training). The multi-epoch results on the last column is obtained by training with the same number of effective epochs with cosine learning rate decay (however, it requires saving the full dataset to storage).

Effective Epochs	Computation	Storage (Prioritized Replay) 1 %	5 %	10 %	Multi-epoch 100% Storage
2	$2/90 \approx 2.2\%$	44.7	45.1	45.7	46.1
4	$4/90 \approx 4.4\%$	55.5	57.1	57.2	59.0
6	$6/90 \approx 6.7\%$	58.9	61.3	62.2	64.1
9	$9/90 = 10\%$	59.3	63.2	65.0	68.2

cross-entropy $\ell(x, y; \theta) = -\log f_y(x; \theta)$ it can also be shown that the priority value simplifies to $P(x, y) = 1 - \alpha f_y(x; \theta)$, which is proportional to the model's confidence on the ground truth label.

3.3 Importance weight

The examples sampled from the priority replay buffer changes the data distribution. To simplify the notation, we omit the label y from the equations in this section. Let $p(x)$ be the distribution of the original data and $q(x)$ be the distribution of data sampled from the replay buffer. The original objective is $\mathbb{E}_p[\ell(x; \theta)]$. Supposing k mini-batches are sampled from the replay buffer at each step, directly combining replay examples and current examples will lead to a minimization of $\mathbb{E}_p[\ell(x; \theta)] + k\mathbb{E}_q[\ell(x; \theta)]$. In order to correct the distribution shift, we use an importance weight $w(x) = p(x)/q(x)$ to each replay example because $\mathbb{E}_q[w(x)\ell(x; \theta)] = \mathbb{E}_p[\ell(x; \theta)]$. Given that the original distribution can be assumed uniform ($p(x) = 1$), the importance weight of each replay example is $w(x) \propto 1/P(x)$, inversely proportional to its priority value. The weights of each mini-batch are normalized to mean 1.

4 Experiments

4.1 Experimental setup

The model in our study is a Residual Neural Network [2] with 50 layers (*aka.* ResNet-50). We use cosine learning rate decay for all experiments, with initial learning rate 0.1. The model is optimized using stochastic gradient descent with Nesterov momentum 0.9. The batch size is 128. We evaluate different approaches with 10 different random orders of the data.

4.2 Results

The results are shown in Table 2. We performed 9 experiments with the replay steps being $k \in \{1, 3, 5, 8\}$ and the size of replay buffer being 1%, 5% and 10% of the dataset. The effective epoch is computed as adding the number of replay steps k by 1 because at receiving each new mini-batch, k mini-batches of the same size are sampled from the replay buffer. So effectively, $k + 1$ back-propagation operations are performed when the model sees a new mini-batch. In addition to the prioritized replay One-Pass solutions, we also show the results of multi-epoch training with the same number of effective epochs. The learning rate decay is adjusted accordingly to decay to minimum at the end of the corresponding epoch. We observe multiple trends from this table:

1. Prioritized replay with 10% memory size achieves performance very close to the multi-epoch training method under the same computational cost. And the multi-epoch method utilizes the full dataset which requires a large data storage. While it is unknown whether multi-epoch gives a performance upper bound, we believe it is a strong target performance to reference.
2. 1% data storage gives a strong starting point for prioritized replay. Having a 1% data storage (equivalent to 100 mini-batches) dramatically improves the naive One-Pass performance by 28.7%. According to the table, the accuracy increase from 1% to 10% storage is 1.0% for 2 epochs, 1.7% for 4 epochs, 3.3% for 6 epochs and 5.7% for 9 epochs.

3. When the buffer size becomes bigger, the accuracy gains are larger when the number of replay steps is larger. From 5% size to 10% size, the model accuracy increase by 0.6% and 0.1% respectively for replay step 1 and 3, while the accuracy increases by 0.9% for replay step 5. The model accuracy saturates quickly if one only increases either storage size or the replay steps. Increasing both of them could potentially incur much bigger accuracy boost.

4.3 Discussion

Priority function. Like recent literature suggested [4, 16], we also observe competitive results using vanilla replay (with a uniform sample memory) in the One-Pass ImageNet problem. Specifically, a uniform memory leads to 64.7% top-1 accuracy with 8 replay steps and 10% storage, slightly worse than the prioritized replay method (65.0%). The standard errors of mean for both are 0.07%. Although the improvement is small, we believe the potential of prioritized replay is not fully explored in this problem and we leave the study of a better designed prioritization as future research.

Importance weights. To study the effectiveness of importance weights applied to the loss function, we evaluate the prioritized replay without importance weight at 1% storage. With 5 replay steps (6 effective epochs), no importance weight results in 58.0% top-1 accuracy, 0.9% lower than the one with importance weight. With 8 replay steps, no importance weight results in 58.3% accuracy, 1.0% lower than using importance weight. The reason of this accuracy gap is due to the fact that the distribution of examples in the replay buffer differs from that of the evaluation data set.

Priority updating. An interesting and somewhat surprising result we obtained is that updating the priorities of the examples retrieved from the replay buffer does not lead to better accuracy, despite the increase in computational cost due to the priority update. An experiment using 5 replay steps and 1% storage shows that, after an example is replayed from the buffer, immediately updating its priority in the replay buffer using the latest model parameters results in a top-1 accuracy of 58.8% while we can achieve 58.9% without updating.

5 Related Work

The OPIN problem is related to continual learning [7, 17–22] and incremental learning [23]. Existing continual learning methods are often evaluated in benchmarks composed of standard supervised learning datasets such as MNIST and CIFAR [11, 24, 25]. Recent efforts are exerted on benchmarks built upon large scale datasets such as ImageNet [26] and Taskonomy [4]. Class-incremental ImageNet [26] is probably the closest to our benchmark in continual learning which splits ImageNet into multiple tasks and each tasks consist of examples labeled as a certain subset of classes. The data for each task is given to the model at once, so the model can train with multiple passes on data in the same task. OPIN differs in that it has no task concept and the data obeys a natural order instead of a manually designed class incremental ordering.

The OPIN problem is also related to stream data learning [27–29] and online learning [30–33]. While there have been many benchmarks utilizing real world streaming data [28], they often have smaller scale, less complex features and limited number of classes. Their solutions are often limited to linear or convex models. The One-Pass ImageNet problem could also contribute to the research of stream data learning from the perspective of deep neural architectures and large scale high dimensional data.

6 Conclusion

We presented the One-Pass ImageNet (OPIN) problem which aims at studying not only the predictive accuracy along with the space and compute efficiency of a supervised learning algorithm. The problem is motivated from real world applications that comes with a stream of extremely large scale data, which leads to impracticality of saving all the data into a dedicate storage and iterating the data for many epochs. We proposed an Error-Prioritized Replay baseline to this problem which achieves 65.0% top-1 accuracy while reducing the required data storage by 90% and reducing the total gradient update compute by 90%, compared against the popular 90-epoch training procedure in ImageNet. We hope OPIN could inspire future research on improving resource efficiency in supervised learning.

Acknowledgement. The authors would like to thank Razvan Pascanu, Murray Shanahan, Eren Sezener, Jianan Wang, Talfan Evans, Sam Smith, Soham De, Yazhe Li, Amal Rannen-Triki, Sven Gowal, Dong Yin, Arslan Chaudhry, Mehrdad Farajtabar, Timothy Nguyen, Nevena Lazic, Iman Mirzadeh, Timothy Mann and John Maggs for their meaningful discussions and contributions.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[2] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[3] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification, 2021.

[4] Yogesh Balaji, Mehrdad Farajtabar, Dong Yin, Alex Mott, and Ang Li. The effectiveness of memory replay in large scale continual learning. In *CVPR Workshop on Continual Learning*, 2021.

[5] Zhipeng Cai, Ozan Sener, and Vladlen Koltun. Online continual learning with natural distribution shifts: An empirical study with visual data. *International Conference on Computer Vision (ICCV)*, 2021.

[6] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] Polina Kirichenko, Mehrdad Farajtabar, Dushyant Rao, Balaji Lakshminarayanan, Nir Levine, Ang Li, Huiyi Hu, Andrew Gordon Wilson, and Razvan Pascanu. Task-agnostic continual learning with hybrid probabilistic models. *ICML 2021 Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021.

[8] Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. Rethinking experience replay: a bag of tricks for continual learning. *International Conference on Pattern Recognition (ICPR)*, 2020.

[9] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[10] Jihwan Bang, Heesu Kim, Youngjoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. In *NeurIPS Continual learning Workshop*, 2018.

[12] Jeffrey S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, March 1985.

[13] Charu C. Aggarwal. On biased reservoir sampling in the presence of stream evolution. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, VLDB '06, page 607–618. VLDB Endowment, 2006.

[14] Miles Osborne, Ashwin Lall, and Benjamin Van Durme. Exponential reservoir sampling for streaming language models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 687–692, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[15] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *International Conference on Learning Representations (ICLR)*, 2016.

[16] Arslan Chaudhry, Albert Gordo, Puneet K Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[17] Dong Yin, Mehrdad Farajtabar, Ang Li, Nir Levine, and Alex Mott. Optimization and generalization of regularization-based continual learning: a loss approximation viewpoint. *ICML Workshop on Continual Learning*, 2020.

[18] Zalán Borsos, Mojmír Mutnỳ, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[19] David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. *32nd AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[20] Gido M van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):1–14, 2020.

[21] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. *International Conference on Learning Representations (ICLR)*, 2019.

[22] 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.

[23] Francisco M. Castro, Manuel J. Marin-Jimenez, Nicolas Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[24] 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*, 2017.

[25] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[26] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019.

[27] Heitor Murilo Gomes, Jesse Read, Albert Bifet, Jean Paul Barddal, and João Gama. Machine learning for streaming data: state of the art, challenges, and opportunities. *SIGKDD Explor.*, 21(2):6–22, 2019.

[28] Vinicius M. A. Souza, Denis M. dos Reis, André Gustavo Maletzke, and Gustavo E. A. P. A. Batista. Challenges in benchmarking stream learning algorithms with real-world data. *Data Mining and Knowledge Discovery*, pages 1 – 54, 2020.

[29] Aymeric Dieuleveut and Francis Bach. Non-parametric stochastic approximation with large step sizes, 2016.

[30] N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Trans. Inf. Theor.*, 50(9):2050–2057, September 2006.

[31] Pierre Tarrès and Yuan Yao. Online learning as stochastic approximation of regularization paths: Optimality and almost-sure convergence. *IEEE Transactions on Information Theory*, 60(9):5716–5735, 2014.

[32] Yiming Ying and Massimiliano Pontil. Online gradient descent learning algorithms. *Foundations of Computational Mathematics*, 8:561–596, 01 2008.

[33] Gunshi Gupta, Karmesh Yadav, and Liam Paull. Look-ahead meta learning for continual learning. *Advances in Neural Information Processing Systems*, 33, 2020.