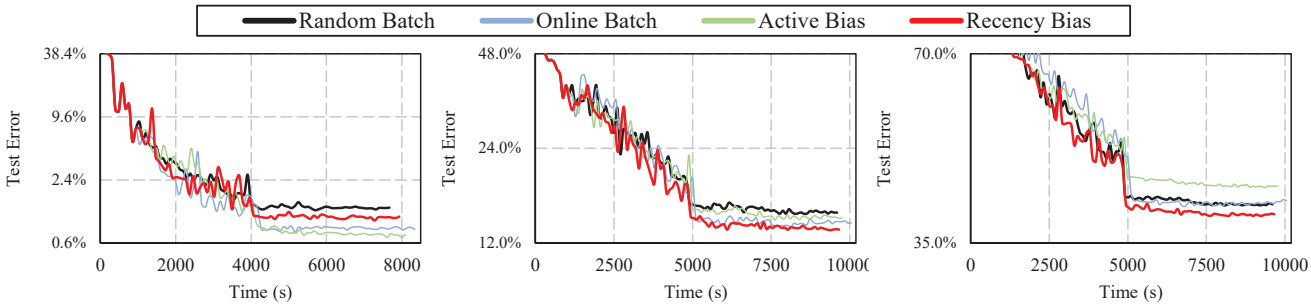
[30] Hwanjun Song, Sundong Kim, Minseok Kim, and Jae-Gil Lee. 2020. Ada-Boundary: Accelerating the DNN training via adaptive boundary batch selection. *Machine Learning* (2020).
[31] Luis Torgo. 2011. *Data mining with R: Learning with case studies*. Chapman and Hall/CRC.
[32] Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. 2016. Learning the curriculum with Bayesian optimization for task-specific word representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 130–139.
[33] Shengjie Wang, Wenruo Bai, Chandrashekhar Lavania, and Jeff Bilmes. 2019. Fixing mini-batch sequences with hierarchical robust partitioning. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*. 3352–3361.
[34] Bernard Widrow, Istvan Kollar, and Ming-Chang Liu. 1996. Statistical theory of quantization. *IEEE Transactions on Instrumentation and Measurement* 45, 2 (1996), 353–361.
[35] Peilin Zhao and Tong Zhang. 2015. Stochastic optimization with importance sampling for regularized loss minimization. In *Proceedings of the 32rd International Conference on Machine Learning*. 1–9.
[36] Tianyi Zhou and Jeff Bilmes. 2018. Minimax curriculum learning: Machine teaching with desirable difficulties and scheduled diversity. In *Proceedings of the 6th International Conference on Learning Representation*.

A GENERALIZABILITY OF *RECENCY BIAS*

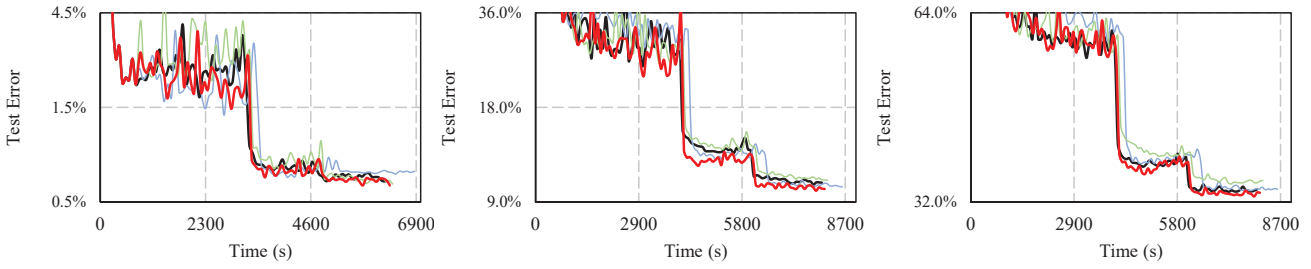
Figures 8, 9, and 10 show the convergence curves of test error for the four batch selection strategies using "DenseNet and an SGD optimizer" (see the right side of Table 2), "ResNet and a momentum optimizer" (see the left side of Table 3), and "ResNet and an SGD optimizer" (see the right side of Table 3), respectively.

The performance dominance of *Recency Bias* was generally consistent regardless of an optimizer and a network architecture. Except in MNIST with an SGD optimizer (Figure 8(a) and Figure 10(a)), *Recency Bias* achieved a significant reduction in test error of 2.32%–19.51% compared with *Random Batch*, 2.51%–8.96% compared with *Online Batch*, and 5.15%–14.64% compared with *Active Bias*, respectively. Therefore, based on this confirmed generalizability, we expect that *Recency Bias* can be smoothly applied for a new emerging optimizer and/or network architecture.



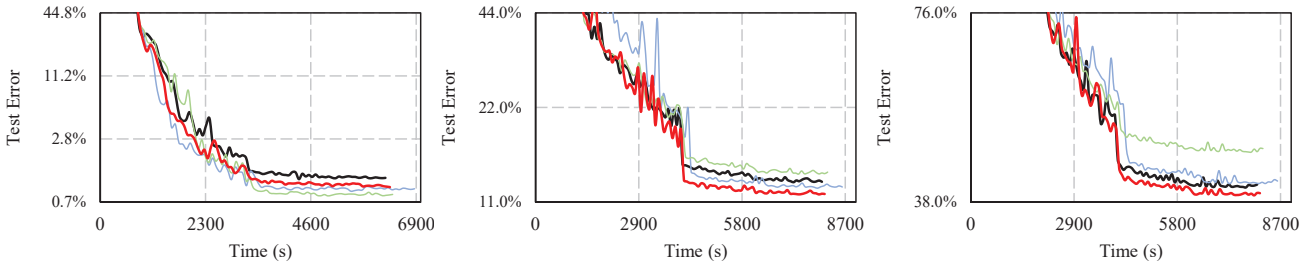
(a) MNIST Test Error. (b) CIFAR-10 Test Error. (c) CIFAR-100 Test Error.

Figure 8: Convergence curves of four batch selection strategies using "DenseNet with SGD" (log-scale).



(a) MNIST Test Error. (b) CIFAR-10 Test Error. (c) CIFAR-100 Test Error.

Figure 9: Convergence curves of four batch selection strategies using "ResNet with momentum" (log-scale).



(a) MNIST Test Error. (b) CIFAR-10 Test Error. (c) CIFAR-100 Test Error.

Figure 10: Convergence curves of four batch selection strategies using "ResNet with SGD" (log-scale).