ConfBatch: Conformal Prediction for Adaptive Batch Sizing

Anna Chatzipapadopoulou ANN.CHATZIPAPADOPOUL@AUEB.GR
Department of Informatics, Athens University of Economics and Business, Athens, Greece
Archimedes Research Unit, Athena Research Center, Athens, Greece

Stavros Toumpis TOUMPIS@AUEB.GR
Department of Informatics, Athens University of Economics and Business, Athens, Greece

John Pavlopoulos IPAVLOPOULOS@AUEB.GR
Department of Informatics, Athens University of Economics and Business, Athens, Greece
Archimedes Research Unit, Athena Research Center, Athens, Greece

Editor: Ernst Ahlberg, Ulf Johansson, Henrik Boström, Alberto Carlevaro, Johan Hallberg Szabadváry and Lars Carlsson

Abstract

We study whether conformal uncertainty can guide adaptive batch size selection during neural network training. We propose *ConfBatch* that periodically computes conformal prediction sets on held-out data, using their size as an uncertainty score to adjust batch size. On CIFAR-10, the opposite-direction ConfBatch variants achieve the highest accuracies, while standard ConfBatch has the lowest wall time among adaptive batch-size methods.

Keywords: conformal prediction, adaptive batch size, uncertainty quantification

1. Idea

Batch size affects both training speed and the noise of stochastic-gradient estimates. Small batches produce noisier updates, whereas large batches are more stable but may use more computation than necessary. The gradient-noise scale can guide batch-size selection (McCandlish et al., 2018), but it relies on gradient statistics.

ConfBatch instead uses conformal prediction sets as an uncertainty signal. At selected epochs, it computes sets $C_t(x_i)$ on held-out data: large sets indicate that several classes remain plausible, whereas singleton sets indicate confident predictions in a classification task. Let s_t be the fraction of singleton sets and K the number of classes. We define

$$U_t = \frac{1}{2}\frac{n^{-1}\sum_{i=1}^{n}\log_2|C_t(x_i)|}{\log_2 K} + \frac{1}{2}(1-s_t).$$

The first term increases with the average set size, while the logarithm prevents very large sets from dominating. Dividing by $\log_2 K$ normalizes it to the number of classes. The second term is small when most predictions are singletons and large when non-singleton sets are common. Thus, larger U_t means greater predictive uncertainty.

Standard ConfBatch compares the current uncertainty U_t with its initial value U_0:

$$B_t^\star = \mathrm{clip}\Big(B_{\min}2^{\eta[U_0-U_t]_+}, B_{\min}, B_{\max}\Big).$$

When uncertainty has not decreased, $[U_0 - U_t]_+ = 0$ and the target remains $B_{\min}$. As the model becomes more confident, the exponent grows and the controller gradually assigns

Table 1: Results on CIFAR-10 with ResNet-18. Values are mean±standard deviation over $N = 5$ seeds. E_{99} and t_{99} denote the first epoch and corresponding wall-clock time at which 99% of the best validation accuracy is reached.

Method	N	Test Acc. (%)	Macro-F1 (%)	Coverage (%)	E_{99}	t_{99} (min)	Wall (min)	FLOPs (PF)	Mean B	Updates (k)
Fixed-small	5	93.97 ± 0.16	93.96 ± 0.16	93.97 ± 0.16	153.6 ± 2.6	39.29 ± 3.10	51.39 ± 4.32	9.50 ± 0.00	64.0 ± 0.0	125.0 ± 0.0
Fixed-large	5	92.40 ± 0.69	92.39 ± 0.69	92.40 ± 0.69	103.4 ± 1.5	22.62 ± 2.13	44.09 ± 4.79	9.50 ± 0.00	1000.0 ± 0.0	8.0 ± 0.0
AdaBatch	5	91.67 ± 0.47	91.67 ± 0.46	91.67 ± 0.47	151.2 ± 0.8	38.62 ± 4.21	49.19 ± 5.03	9.50 ± 0.00	93.5 ± 0.0	85.5 ± 0.0
Smith	5	93.53 ± 0.23	93.53 ± 0.23	93.53 ± 0.23	151.2 ± 0.4	36.47 ± 3.11	47.06 ± 4.15	9.50 ± 0.00	115.3 ± 0.0	69.4 ± 0.0
CABS	5	88.50 ± 0.19	88.48 ± 0.19	89.67 ± 0.48	44.6 ± 5.9	35.65 ± 6.82	163.31 ± 10.12	9.48 ± 0.00	254.0 ± 5.2	31.51 ± 0.65
SimiGrad	5	93.76 ± 0.27	93.75 ± 0.27	93.76 ± 0.27	106.0 ± 2.5	32.23 ± 2.30	53.32 ± 4.04	9.50 ± 0.00	713.6 ± 40.7	11.24 ± 0.66
ConfBatch	5	94.16 ± 0.30	94.16 ± 0.30	94.16 ± 0.30	120.6 ± 22.3	27.94 ± 5.19	45.85 ± 0.96	10.06 ± 0.00	135.7 ± 23.1	60.27 ± 9.85
ConfBatch-opposite	5	94.38 ± 0.26	94.38 ± 0.26	94.38 ± 0.26	117.2 ± 20.6	27.43 ± 5.10	47.25 ± 4.27	10.06 ± 0.00	137.4 ± 5.4	58.28 ± 2.17
ConfBatch-aggressive	5	93.13 ± 0.59	93.13 ± 0.57	93.13 ± 0.59	141.0 ± 1.6	44.64 ± 7.81	60.25 ± 7.91	10.05 ± 0.00	1906.1 ± 274.2	4.27 ± 0.60
ConfBatch-opposite-aggressive	5	$\mathbf{94.40} \pm 0.30$	94.40 ± 0.30	94.40 ± 0.30	175.4 ± 2.3	43.21 ± 1.13	49.51 ± 0.82	10.05 ± 0.00	66.2 ± 0.4	120.81 ± 0.73

larger batches. The parameter η controls the size of the batch adjustment, while clipping keeps it within $[B_{\min}, B_{\max}]$. We also evaluate the opposite-direction controller:

$$B^{\star}_{t,\mathrm{opp}} = \mathrm{clip}\Big(B_a 2^{\eta(U_t - U_0)}, B_{\min}, B_{\max}\Big),$$

which increases the target batch size when uncertainty is higher than the reference level, decreasing it when it is lower. Both smooth and limit batch-size changes, while a coverage safeguard prevents increases when empirical conformal coverage may be below target.

2. Results and Discussion

Evaluation. We train ResNet-18 on CIFAR-10 for 200 epochs and report the mean and st. deviation over five seeds. We compare ConfBatch with fixed-batch training, AdaBatch (Devarakonda et al., 2017), Smith's schedule (Smith et al., 2017), CABS (Balles et al., 2016), and SimiGrad (Qin et al., 2021). Besides predictive performance and coverage, we measure wall time, estimated FLOPs, mean batch size, and optimizer updates. E_{99} and t_{99} are the first epoch and corresponding wall-clock time at which 99% of the best validation accuracy is reached, measuring how quickly each method approaches its best performance. We use $\alpha = 0.1$, corresponding to 90% coverage.

Results and Future Work. The two opposite-direction ConfBatch variants achieve the highest test accuracies; ConfBatch-opposite-aggressive performs best at $94.40 \pm 0.30\%$, followed by ConfBatch-opposite at $94.38 \pm 0.26\%$. Standard ConfBatch remains competitive at $94.16 \pm 0.30\%$ and has the second-fastest total wall time among all methods (45.85 ± 0.96 minutes), after fixed-large, making it the fastest adaptive batch-size method. ConfBatch-opposite similarly obtains the second-lowest t_{99} overall (27.43 ± 5.10 minutes) and the lowest among adaptive methods. Although fixed-large is faster, its test accuracy falls to $92.40 \pm 0.69\%$. Future work will evaluate ConfBatch on additional datasets and neural-network architectures.

Acknowledgments

Work supported by the EU (AIRIS, Grant No. 101289094), Greece 2.0/NextGenerationEU (MIS 5154714), and RC/AUEB (project 11387901).

References

Lukas Balles, Javier Romero, and Philipp Hennig. Coupling adaptive batch sizes with learning rates. *arXiv preprint arXiv:1612.05086*, 2016.

Aditya Devarakonda, Maxim Naumov, and Michael Garland. Adabatch: Adaptive batch sizes for training deep neural networks. *arXiv preprint arXiv:1712.02029*, 2017.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

Heyang Qin, Samyam Rajbhandari, Olatunji Ruwase, Feng Yan, Lei Yang, and Yuxiong He. Simigrad: Fine-grained adaptive batching for large scale training using gradient similarity measurement. *Advances in Neural Information Processing Systems*, 34:20531–20544, 2021.

Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.