Faster Training by Automatically Selecting the Best Training Data for Computer Vision Tasks

Tony Cai, Dennis Duan, and Ananth Agarwal

6/4/2022

Stanford University

Motivation: Increasingly Expensive Model Training

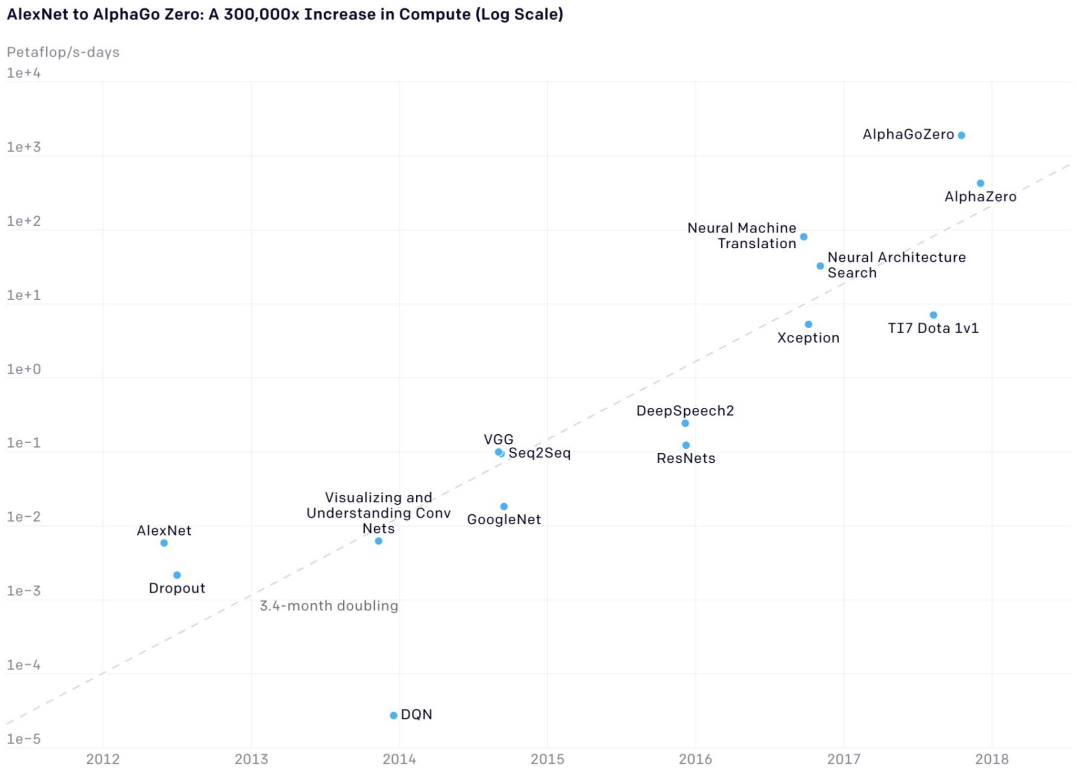
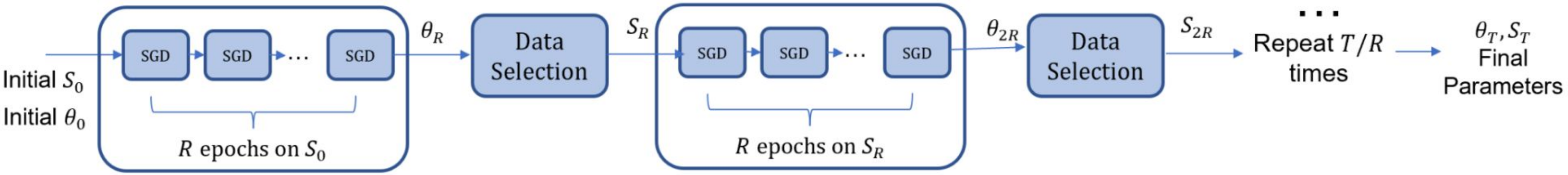


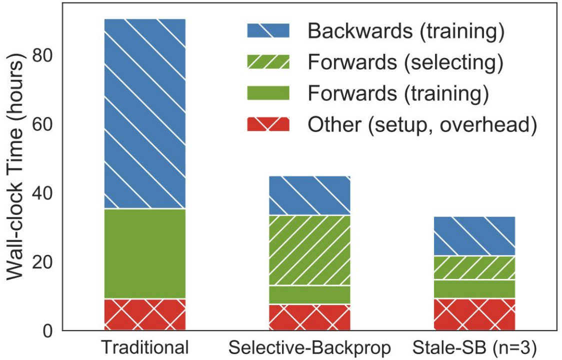
Image Credit: Amodei, Dario, and Danny Hernandez. “AI and Compute.” OpenAI, OpenAI, 16 May 2018, https://openai.com/blog/ai-and-compute/.

Previous Approaches

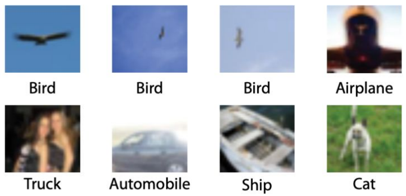
- Grad-Match: Periodically choose subsets of training data to train on



- Selective-Backprop: Only perform backpropagation on high-loss examples



(a) Examples chosen least frequently by SB



(b) Examples chosen most frequently by SB

Image Credits:
Jiang, A. H., Wong, D. L. K., Zhou, G., Andersen, D. G., Dean, J., Ganger, G. R., ... & Pillai, P. (2019). Accelerating deep learning by focusing on the biggest losers. arXiv preprint arXiv:1910.00762.
Killamsetty, K., Durga, S., Ramakrishnan, G., De, A., & Iyer, R. (2021, July). Grad-match: Gradient matching based data subset selection for efficient deep model training. In International Conference on Machine Learning (pp. 5464-5474). PMLR.

Our Contributions

- Key observation: Most often, model training occurs multiple times, for example for hyperparameter and model search

- Can we use training statistics from prior training runs to help train models on the same dataset more efficiently with minimal loss in performance?

Our Contributions

- For the image classification task:
  - Reproduce the training speedups reported by Grad-Match on the standard CIFAR-10 dataset, evaluate it against a new random selection baseline, and propose and evaluate a variant that reuses subsets chosen from previous training runs.

- For the object detection task:
  - Evaluate if the benefits of Selective-Backprop translate to object detection on the popular MS COCO dataset

Image Classification - Methods

- Reproduce the training speedups reported by Grad-Match on the standard CIFAR-10 dataset, **evaluate it against a new random selection baseline**, and propose and evaluate a variant that reuses subsets chosen from previous training runs.
    - Random subset selection: choose random examples to fill budget
    - **RandomPB**: divide train set into fixed batches, then randomly choose batches

Image Classification - Methods

- Reproduce the training speedups reported by Grad-Match on the standard CIFAR-10 dataset, evaluate it against a new random selection baseline, and **propose and evaluate a variant that reuses subsets chosen from previous training runs**.
  - **Cached-Grad-Match**: run an initial training run with Grad-Match, and on subsequent runs “replay” the same subsets selected
    - Saves computation time, but how robust is it?

Image Classification - Experiments

- Cached-Grad-Match
  - Reuse Grad-Match’s data selection on ResNet training for different learning rates
  - Reuse Grad-Match’s data selection on ResNet training for optimizers
  - Reuse Grad-Match’s data selection on MobileNetV2 training for ResNet training
- Baseline
  - RandomPB
  - Training on full dataset

Image Classification - Fixed Accuracy Speedup

	Speedup		
Learning rate	0.001	0.003	0.03
GRAD-MATCH	0.81	0.82	2.72
CACHED-GRAD-MATCH	**1.00**	**0.99**	**7.92**

Table 1. Fixed-accuracy speedups over full training using different learning rates.

	Speedup		
	Adam	RMSProp	SGD
GRAD-MATCH	**0.83**	0.81	3.07
CACHED-GRAD-MATCH	0.67	**0.97**	**4.68**

Table 2. Fixed-accuracy speedups over full training using different optimizers.

	Speedup on ResNet
GRAD-MATCH	3.86
GRAD-MATCH-warm	**4.69**
CACHED-GRAD-MATCH	2.24
CACHED-GRAD-MATCH-warm	4.51

Table 3. Fixed-accuracy speedups over full training for different model architectures.

Image Classification - Training Speed vs Error

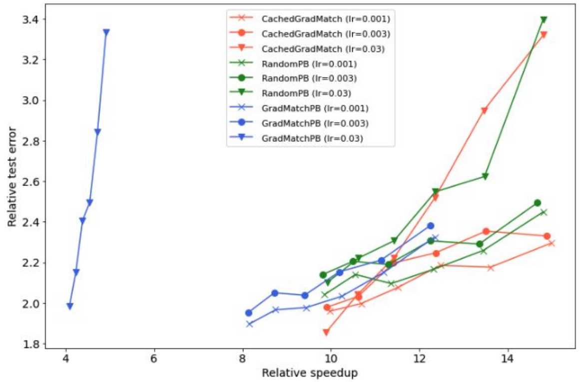


Figure 1. Speedup-accuracy tradeoffs of subset selection algorithms using different learning rates.

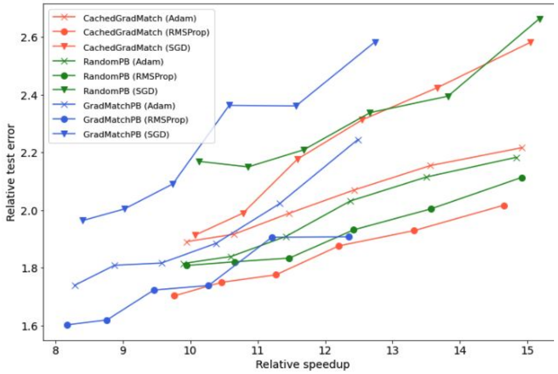


Figure 2. Speedup-accuracy tradeoffs of subset selection alg rithms using different optimizers.

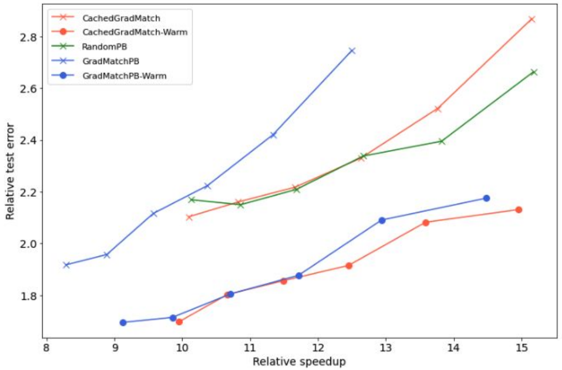
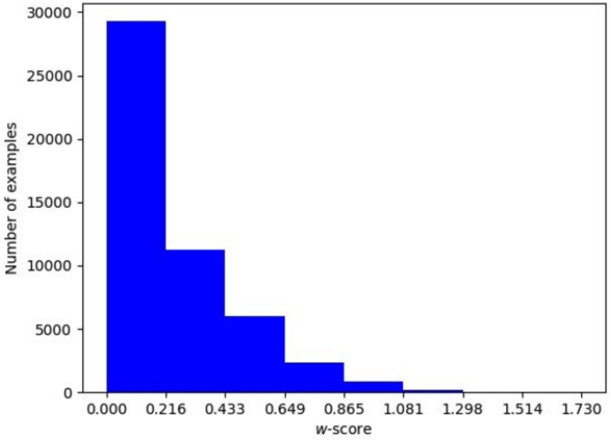


Figure 3. Speedup-accuracy tradeoffs of subset selection algorithms for different model architectures.

Image Classification - Results

Low-priority images

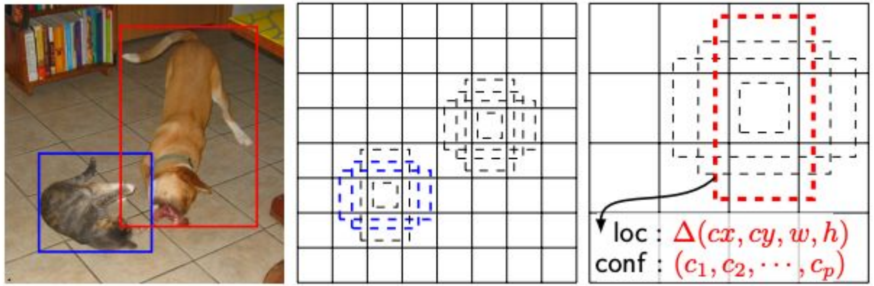
High-priority images



Priority histogram

Object Detection - Methods

- Single Shot Multibox Detector (SSD)
  - ResNet backbone + conv layers + classification, regression heads



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

- Loss Function

$$L_{obj}(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

Image Credit: Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy,
Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. CoRR, abs/1512.02325, 2015.

Object Detection - Methods

- Selective-Backprop
    - Combine Jiang et al. original approach with MosaicML suggestions

- Nvidia SSD PyTorch library: ResNet backbone, input size 300x300
    - Integrates with COCO's Python library

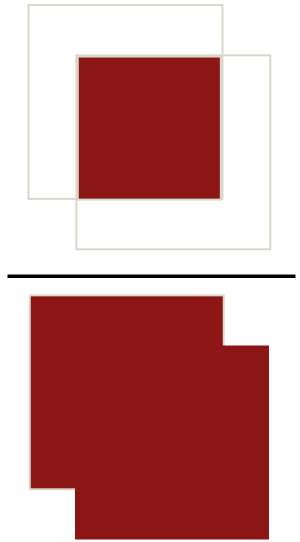
Algorithm 1 SELECTIVE-BACKPROP for Object Detection

```
for epoch in range(epochs) do
    for i, X_{b_size} in enumerate(data_loader) do
        if epoch in [sb_start, sb_end] then
            losses = forward(X_{b_size})
            if i % interrupt == 0 then
                backward(losses)
            else
                X_s = P_select(losses)
                backward(forward(X_s))
            end if
        else
            backward(forward(X_{b_size}))
        end if
    end for
end for
```

Object Detection - Methods

- Evaluation Metrics
  - Total Training Time
  - Mean average precision: mAP [0.50:0.95]
    - All object sizes
    - Small
    - Medium
    - Large

- Intersection over Union (IoU):



Object Detection - Experiments

1. Baseline: No Selective-Backprop
2. Selective-Backprop, interrupt = 2
3. Selective-Backprop, interrupt = 3

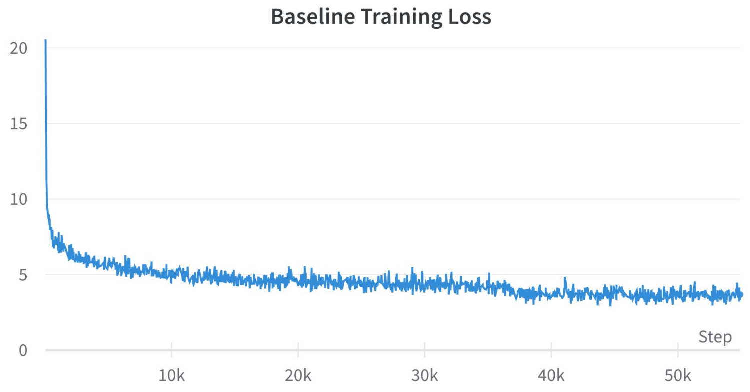
- Common parameters:
    - Batch size = 32
    - Epochs = 30

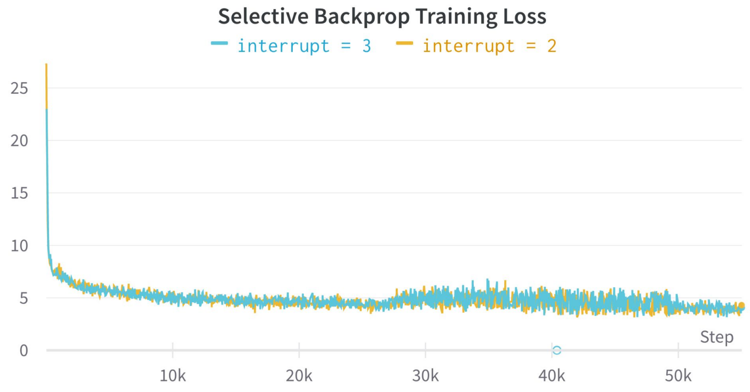
- Selective-Backprop (variables defined in Alg. 1)
    - Start (sb_start) = 0.5
    - End (sb_end) = 0.9
    - Keep (s) = 0.5

Object Detection - Results

	mAP IoU 0.50:0.95				Time	
	All	Small	Med.	Large	Hours	t_{diff} (s)
Base	**9.68**	**2.65**	**10.85**	**15.48**	7.61	N/A
SB_2	9.09	2.47	9.71	14.8	**7.44**	41.07
SB_3	8.79	2.22	9.53	14.1	N/A[1]	**51.14**

$$t_{diff} = \underset{\substack{i \in [1,\texttt{epochs}] \\ i \notin [\texttt{sb\_start},\texttt{sb\_end})}}{\bar{t_i}} - \underset{\substack{j \in [1,\texttt{epochs}] \\ j \in [\texttt{sb\_start},\texttt{sb\_end})}}{\bar{t_j}}$$



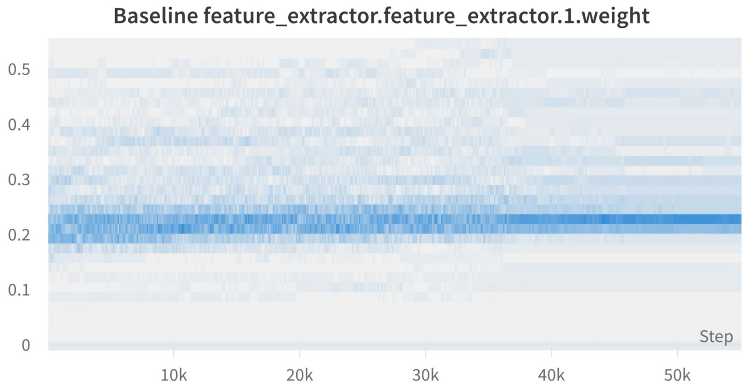


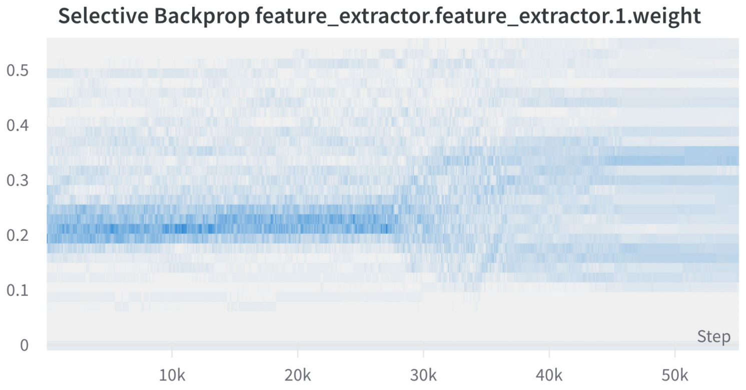
[1] The AWS EC2 instance was stopped and restarted between running SB3 and the baseline and SB2, so the SB3 absolute training time is not directly comparable to the other two due to potential GPU environment differences.

Object Detection - Discussion

- Speedup, but accuracy lost

- Alg. 1 has a visible effect on model parameters in one layer of the ResNet50 backbone:





Future Work

- Image Classification
    - Convergence analysis
    - Other data selection heuristics
- Object Detection
    - Multiple re-runs, Hyperparameter sweeps
    - Larger batch size
    - Grad-Match and Cached-Grad-Match

Conclusion

- Found Grad-Match speedup for image classification, but its performance varies substantially with different hyperparameters

- Reusing prior training (Cached-Grad-Match) shows performance gains without extra cost of adaptive data selection

- Benefit of data selection is less pronounced for object detection

References

Amodei, Dario, and Danny Hernandez. “AI and Compute.” OpenAI, OpenAI, 16 May 2018, https://openai.com/blog/ai-and-compute/.

Jiang, A. H., Wong, D. L. K., Zhou, G., Andersen, D. G., Dean, J., Ganger, G. R., ... & Pillai, P. (2019). Accelerating deep learning by focusing on the biggest losers. arXiv preprint arXiv:1910.00762.

Krishnateja Killamsetty, Dheeraj Bhat, Ganesh Ramakrishnan, and Rishabh Iyer. CORDS: COResets and Data Subset selection for Efficient Learning, 3 2022.

Killamsetty, K., Durga, S., Ramakrishnan, G., De, A., & Iyer, R. (2021, July). Grad-match: Gradient matching based data subset selection for efficient deep model training. In International Conference on Machine Learning (pp. 5464-5474). PMLR.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. CoRR, abs/1405.0312, 2014.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy,
Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. CoRR, abs/1512.02325, 2015.

Nvidia. SSD300 v1.1 For PyTorch, 3 2019.

Abhinav Venigalla. MosaicML: Selective Backprop, 2021.