SELECTION VIA PROXY: INCREASING THE COMPUTATIONAL EFFICIENCY OF DEEP ACTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Active learning techniques can improve data efficiency of labeling but can be computationally expensive to apply in deep learning. Unlike in other areas of machine learning, the feature representations that these techniques depend on are learned in deep learning rather than given, requiring substantial training times. In this work, we show that we can greatly improve the computational efficiency of deep active learning by using a small proxy model to select which data points to label. By removing hidden layers from the target model, using smaller architectures, or training for fewer epochs, we create proxies that are an order of magnitude faster to train. Although these small proxy models have higher error rates, we find that they empirically provide useful signal for data selection. We evaluate this "selection via proxy" (SVP) approach with two active learning methods across five datasets: CIFAR10, CIFAR100, ImageNet, Amazon Review Polarity, and Amazon Review Full. Applying SVP to active learning can give an order of magnitude improvement in data selection runtime (i.e., the time it takes to repeatedly train and select points) without significantly increasing the final error.

1 INTRODUCTION

Active learning improves the *data efficiency* of machine learning by identifying the most informative training examples. To quantify informativeness, these methods depend on semantically meaningful features or a trained model to calculate uncertainty. Concretely, active learning selects points to label from a large pool of unlabeled data by repeatedly training a model on a small pool of labeled data and selecting additional examples to label based on the model's uncertainty (e.g., the entropy of predicted class probabilities) or other heuristics (Settles, 2011; 2012; Lewis & Gale, 1994).

Unfortunately, classical active learning methods are often prohibitively expensive to apply in deep learning (Shen et al., 2017; Sener & Savarese, 2018; Kirsch et al., 2019). Unlike other machine learning methods, deep learning models learn complex internal semantic representations (hidden layers) from raw inputs (e.g., pixels or characters) that enable them to achieve state-of-the-art performance but result in substantial training times. Many active learning techniques require this feature representation *before* they can accurately identify informative points. As a result, new deep active learning methods request labels in large batches to avoid retraining the model too many times (Shen et al., 2017; Sener & Savarese, 2018; Kirsch et al., 2019). However, batch active learning still requires training a full deep model for every batch, which is costly for large models.

In this paper, we propose *selection via proxy (SVP)* as a novel way to make existing active learning methods more computationally efficient for deep learning. SVP uses the feature representation from a separate, less computationally intensive proxy model in place of the representation from the much larger and more accurate target model we aim to train. SVP builds on the idea of heterogeneous uncertainty sampling from Lewis & Catlett (1994), which showed that an inexpensive classifier (e.g., naïve Bayes) can select points to label for a much more computationally expensive classifier (e.g., decision tree). In our work, we show that small deep learning models can similarly serve as an inexpensive proxy for data selection in deep learning, significantly accelerating active learning techniques. To create these cheap proxy models, we can scale down deep learning models by removing layers, using smaller model architectures, or training them for fewer epochs. While these scaled-down models achieve significantly lower accuracy than larger models, we surprisingly find that they still provide useful representations to rank and select points (i.e., high Spearman's and Pearson's correlations with much larger models on metrics such as uncertainty (Settles, 2012) and submodular algorithms such as greedy k-centers (Wolf, 2011)). Because these proxy models are quick to train, we can identify which points to select nearly as well as the larger target model but significantly faster.

We empirically evaluated SVP for active learning on five datasets: CIFAR10, CIFAR100 (Krizhevsky & Hinton, 2009), ImageNet (Russakovsky et al., 2015), Amazon Review Polarity, and Amazon Review Full (Zhang et al., 2015). We considered both least confidence uncertainty sampling (Settles, 2012; Shen et al., 2017; Gal et al., 2017) and the core-set approach from Sener & Savarese (2018) with a variety of proxies. Across all datasets, we found that SVP matches the accuracy of the traditional approach of using the same large model for both selecting points and the final prediction task. Depending on the proxy, SVP yielded up to a 7× speed-up on CIFAR10 and CIFAR100, $41.9\times$ speed-up on Amazon Review Polarity and Full, and $2.9\times$ speed-up on ImageNet in data selection runtime (i.e., the time it takes to repeatedly train and select points). For example, the Amazon Review results were achieved using fastText as a proxy for VDCNN29, which takes less than 10 minutes to train instead of 16 hours. These results demonstrate that SVP is a promising, yet simple approach to make active learning methods computationally feasible.

2 METHODS

2.1 ACTIVE LEARNING

Pool-based active learning starts with a large pool of unlabeled data $U = \{\mathbf{x}_i\}_{i\in[n]}$ where $[n] = \{1, \ldots, n\}$. Each example is from the space $\mathcal{X}$ with an unknown label from the label space $\mathcal{Y}$ and is sampled *i.i.d.* over the space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ as $\{\mathbf{x}_i, y_i\} \sim p_{\mathcal{Z}}$. Initially, methods label a small pool of points $s^0 = \{s^0_j \in [n]\}_{j\in[m]}$ chosen uniformly at random. Given U, a loss function ℓ, and the labels $\{y_{s^0_j}\}_{j\in[m]}$ for the initial random subset, the goal of active learning is to select up to a budget of b points from U to label that will minimize the generalization error of a learning algorithm A.

Baseline. In this paper, we applied SVP to least confidence uncertainty sampling (Settles, 2012; Shen et al., 2017; Gal et al., 2017) and the recent core-set approach to active learning from Sener & Savarese (2018). Like recent work for deep active learning (Shen et al., 2017; Sener & Savarese, 2018; Kirsch et al., 2019), we considered a batch setting with K rounds where we selected $\frac{b}{K}$ points in every round. Following Gal et al. (2017); Sener & Savarese (2018); Kirsch et al. (2019), we reinitialized the target model and retrained on all of the labeled data collected over previous rounds (denoted as $A^T_{s^0\cup\ldots\cup s_k}$ or A^T_k) to avoid any correlation between selections (Frankle & Carbin, 2018; Kirsch et al., 2019). Then using A^T_k, we either calculated the model's confidence as:

$$f_{\text{confidence}}(\mathbf{x}; A^T_k) = 1 - \max_{\hat{y}} P(\hat{y}|\mathbf{x}; A^T_k)$$

and selected the examples with the lowest confidence or extracted a feature representation from the model's final hidden layer and computed the distance between examples (i.e., $\Delta(\mathbf{x}_i, \mathbf{x}_j; A^T_k)$) to select points according to the greedy k-centers method from Wolf (2011); Sener & Savarese (2018) (Algorithm 1). The same model was trained on the final b labeled points to yield the final model, A^T_K, which was then tested on a held-out set to evaluate error and quantify the quality of the selected data.

Algorithm 1 GREEDY K-CENTERS (WOLF, 2011; SENER & SAVARESE, 2018)

```
Input: data $\mathbf{x}_i$, existing pool $s^0$, trained model $A^T_0$, and a budget $b$
Initialize $s = s^0$
repeat
    $u = \arg\max_{i\in[n]\setminus s} \min_{j\in s} \Delta\left(\mathbf{x}_i, \mathbf{x}_j; A^T_0\right)$
    $s = s \cup \{u\}$
until $|s| = b + |s^0|$
return $s \setminus s^0$
```

2.2 APPLYING SELECTION VIA PROXY

SVP can be applied by replacing the models used to compute data selection metrics such as uncertainty with proxy models. Specifically, we replaced the model trained at each batch (A^T_k) with a proxy (A^P_k), but then trained the same final model A^T_K once the budget b was reached, as shown in Figure 1. We explored two main methods to create our proxy models:

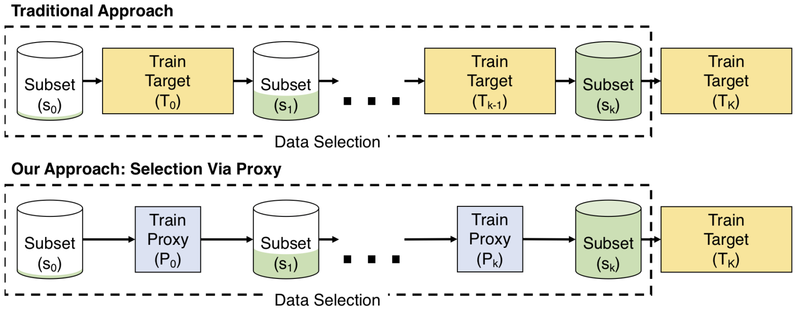

Figure 1: **SVP applied to active learning**. We followed the same iterative procedure of training and selecting points to label as traditional approaches but replaced the target model with a cheaper-to-compute proxy model. Empirically, we found the proxy and target model have high rank-order correlation, leading to similar selections and downstream results.

Scaling down. For deep models with many layers, reducing the dimension or the number of hidden layers reduces training times considerably with only a small drop in accuracy. For example, the accuracy of deep ResNet models only slightly diminishes as layers are dropped from the network (He et al., 2016b;a). A ResNet20 model achieves a top-1 error of 7.6% on CIFAR10 in 26 minutes, while a larger ResNet164 model only reduces error by 2.5%, but takes 4 hours (Figure 2a in the Appendix). Looking across architectures can also substantially reduce computational complexity with only a small increase in error. We exploit these diminishing returns to scale down to a proxy that can be trained quickly but still provides a good approximation of the target's decision boundary.

Training for a fewer epochs. Similarly, a significant amount of training is spent on a relatively small reduction in error. While training ResNet20, almost half of the training time (i.e., 12 minutes out of 26 minutes) is spent on a 1.4% improvement in test error, as shown in Figure 2a in the Appendix. Based on this observation, we also explored training proxy models for a smaller number of epochs.

3 Results

3.1 Experimental Setup

Datasets. Our experiments included three image classification datasets: CIFAR10, CIFAR100 (Krizhevsky & Hinton, 2009), and ImageNet (Russakovsky et al., 2015); and two text classification datasets: Amazon Review Polarity and Full (Zhang et al., 2015). CIFAR10 is a coarse-grained classification task over 10 classes, and CIFAR100 is a fine-grained task with 100 classes. Both datasets contain 50,000 images for training and 10,000 images for testing. ImageNet has 1.28 million training images and 50,000 validation images that belong to 1 of 1,000 classes. Amazon Review Polarity (2-classes) has 3.6 million reviews with an additional 400,000 reviews for testing. Amazon Review Full (5-classes) has 3 million reviews with an additional 650,000 reviews for testing.

Models. For CIFAR10 and CIFAR100, we used ResNet164 with pre-activation from He et al. (2016b) as our large target model. The smaller, proxy models are also ResNet architectures with pre-activation, but they use pairs of 3×3 convolutional layers as their residual unit rather than bottlenecks. For ImageNet, we used the original ResNet architecture from He et al. (2016a) implemented in PyTorch with ResNet50 as the target and ResNet18 as the proxy. For Amazon Review Polarity and Full, we used VDCNN (Conneau et al., 2017) and fastText (Joulin et al., 2016) with VDCNN29 as the target.

3.2 Active Learning

We explored the impact of SVP on two active learning techniques: least confidence uncertainty sampling and the coreset approach from Sener & Savarese (2018). Starting with an initial random subset of 2% of the data, we selected 8% of the remaining unlabeled data for the first round and 10% for subsequent rounds until the labeled data reached the budget b and retrained the models from scratch between rounds as described in Section 2.1. Across datasets, SVP sped up data selection without significantly impacting the final predictive performance of the target.

Table 1: Average (± 1 std.) data selection speed-ups from 3 runs of active learning using least confidence uncertainty sampling with varying proxies and labeling budgets on four datasets. **Bold** speed-ups indicate settings that either achieve lower error or are within 1 std. of the mean top-1 error for the baseline approach of using the same model for selection and the final predictions.

		Data Selection Speed-up				
	Budget (b/n)	10.0%	20.0%	30.0%	40.0%	50.0%
Dataset	**Selection Model**					
CIFAR10	ResNet164 (Baseline)	**1.0**×	**1.0**×	**1.0**×	**1.0**×	**1.0**×
	ResNet110	**1.8**×	**1.9**×	**1.9**×	**1.8**×	**1.8**×
	ResNet56	**2.6**×	**2.9**×	**3.0**×	**3.1**×	3.1×
	ResNet20	**3.8**×	**5.8**×	**6.7**×	**7.0**×	7.2×
CIFAR100	ResNet164 (Baseline)	**1.0**×	**1.0**×	**1.0**×	**1.0**×	**1.0**×
	ResNet110	**1.5**×	**1.6**×	**1.6**×	**1.6**×	1.6×
	ResNet56	**2.4**×	2.7×	**3.0**×	**2.9**×	**3.1**×
	ResNet20	4.0×	**5.8**×	**6.6**×	**7.0**×	**7.2**×
ImageNet	ResNet50 (Baseline)	**1.0**×	**1.0**×	**1.0**×	**1.0**×	**1.0**×
	ResNet18	**1.2**×	**1.3**×	**1.4**×	**1.5**×	**1.6**×
Amazon Review Polarity	VDCNN29 (Baseline)	**1.0**×	**1.0**×	**1.0**×	**1.0**×	**1.0**×
	VDCNN9	**1.9**×	1.8×	**1.8**×	**1.8**×	1.8×
	fastText	10.6×	20.6×	32.2×	**41.9**×	51.3×

CIFAR10 and CIFAR100. For least confidence uncertainty sampling and greedy k-centers, SVP sped-up data selection by up to 7× and 3.8× respectively without impacting data efficiency (see Tables 1 and 2) despite the proxy achieving substantially higher top-1 error than the target ResNet164 model (see Figure 4 in the Appendix). The speed-ups for least confidence were a direct reflection of the difference in training time between the proxy in the target models. As shown in Figures 2 and 3 in the Appendix, ResNet20 was about 8× faster to train than ResNet164, taking 30 minutes to train rather than 4 hours. Larger budgets required more rounds of selection and, in turn, more training, which led to larger speed-ups as training became a more significant fraction of the total time. Training for fewer epochs provided a significant error reduction compared to random sampling but was not as good as training for the full schedule (see Table 3 in the Appendix). For greedy k-centers, the speed-ups increased more slowly because executing the selection algorithm added more overhead.

ImageNet. For least confidence, SVP sped-up data selection by up to 1.6× (Table 1) despite ResNet18's higher error compared to ResNet50 (Figure 4g in the Appendix). Training for fewer epochs increased the speed-up to 2.9× without increasing the error rate of ResNet50 (Table 3). Greedy k-centers was too slow on ImageNet due to the quadratic complexity of Algorithm 1.

Amazon Review Polarity and Amazon Review Full. On Amazon Review Polarity, SVP with a fastText proxy for VDCNN29 led to up to a relative error reduction of 14% over random sampling for large budgets (Table 2), while being up to 41.9× faster at data selection than the baseline approach (Table 1). Despite fastText's architectural simplicity compared to VDCNN29 and higher error (Figure 4e), the calculated confidences signaled which examples would be the most informative. For all budgets, VDCNN9 was within 0.1% top-1 error of VDCNN29, giving a consistent 1.8× speed-up. On Amazon Review Full, neither the baseline least confidence uncertainty sampling approach nor the application of SVP outperformed random sampling (see Table 2 in the Appendix), so the data selection speed-ups were uninteresting even though they were similar to Amazon Review Polarity. For both datasets, greedy k-centers was too slow as mentioned above in the ImageNet experiments.

4 Conclusion

In this work, we introduced selection via proxy (SVP) to improve the computational efficiency of active learning in deep learning by substituting a cheaper proxy model's representation for an expensive model's during data selection. Applied to least confidence uncertainty sampling and Sener & Savarese (2018)'s core-set approach, SVP achieved up to a 41.9× and 3.8× improvement in runtime respectively with no significant increase in error. Our results demonstrate that SVP is a promising approach to reduce the computational requirements of active learning for deep learning.

References

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 1107–1116. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/E17-1104.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1183–1192. JMLR. org, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *arXiv preprint arXiv:1906.08158*, 2019.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pp. 148–156. Elsevier, 1994.

David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 3–12. Springer-Verlag New York, Inc., 1994.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

Burr Settles. From theories to queries: Active learning in practice. In Isabelle Guyon, Gavin Cawley, Gideon Dror, Vincent Lemaire, and Alexander Statnikov (eds.), *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, pp. 1–18, Sardinia, Italy, 16 May 2011. PMLR. URL http://proceedings.mlr.press/v16/settles11a.html.

Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6 (1):1–114, 2012.

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 252–256, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2630. URL https://www.aclweb.org/anthology/W17-2630.

Gert W Wolf. Facility location: concepts, models, algorithms and case studies., 2011.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pp. 649–657, 2015.