Learning Active Learning Policies for Visual Recognition

Viraj Prabhu*
College of Computing
Georgia Institute of Technology
virajp@gatech.edu

Abstract

We consider the problem of learning policies for active learning via reinforcement learning for visual recognition tasks. While traditional active learning employs various heuristics as acquisition functions, we approach this by framing the streaming active learning setting as a Markov Decision Process. At each timestep a decision is to be made of whether or not to query an oracle for a label, which are used to train a model online; an optimal policy leads to training the best classifier for the downstream task. Such policies can both overcome the need to hand-design heuristics by learning data-driven acquisition functions, and further can work well even when training a model from scratch i.e. in a *cold-start* setting. We learn such active learning policies for a digit recognition task, and demonstrate strong performance against baselines. Finally, we study how well such a learned policy can transfer to a (potentially low resource) target domain.

1 Introduction

In recent times, deep convolutional neural networks have shown great promise for visual recognition tasks, achieving impressive gains over previous methods on challenging benchmarks [Krizhevsky et al., 2012, Ren et al., 2015], and receiving widespread adoption in several real-world applications [He et al., 2017, Chen et al., 2018]. However, such networks have well-documented problems of being data-hungry [Zhu et al., 2012, Sun et al., 2017], and require large amounts of labeled examples to train effectively. The real world exhibits significant visual variations, and collecting annotations at scale for each recognition setting is not a feasible solution.

Active learning [Settles, 2009] seeks to reduce this labeling burden by picking the most valuable instances in a large unlabeled set to get labeled by an oracle. This field has seen extensive work and several effective heuristics have been proposed as utility functions, such as picking instances the model is most uncertain about, or the instance whose label is expected to maximally reduce the model's uncertainty about other instances [Bachman et al., 2017]. However, these heuristics have been found to often learn suboptimal and "myopic" policies that do not generalize across datasets and domains. In addition, the "expert features" that several of these heuristics rely on, such as uncertainty and margins, can be uncalibrated when obtained from a model that is being trained via active learning "from scratch" i.e. in what is described as a cold-start setting. In such scenarios, more robust strategies are required that can adapt to varying degrees of model calibration.

Recent work [Fang et al., 2017, Konyushkova et al., 2017, Bachman et al., 2017] has shown the promise of learning end-to-end policies for active learning via meta-learning, that outperform traditional heuristics on certain tasks by learning acquisition functions from data rather than employing engineered features. These policies are typically learned using deep reinforcement learning tech-

*Course project for CS 8803: Adaptive Control and Reinforcement Learning, Spring 2019.

niques, such as deep Q-learning or policy gradient methods. Note that such strategies are naturally amenable to both batch mode and streaming active learning setups.

However, such policy active learning methods have a limitation – since the deep RL methods are often quite sample inefficient, significant amounts of labeled data are required to learn effective policies, especially when training a model from scratch. This potentially defeats the purpose of doing active learning, as if we had large amounts of labeled data for a domain, we wouldn't need to do active learning in the first place! The true utility therefore is in learning a *transferable* policy that can easily be transferred (potentially via finetuning) to a new, potentially low-resource domain.

Contributions. In this work, we make the following contributions:

1. We study the problem of meta-learning active learning policies for *visual recognition tasks*, and study how the performance of such policies compares with traditional active learning heuristics. We do this by posing the active learning setup as a Markov Decision Process and use deep reinforcement learning techniques to learn policies end-to-end.
2. Further, we study if we can efficiently transfer such policies across visual domains. Such effective transfer can be of great practical utility in transferring policies learned on data-rich visual domains to low-resource domains.

2 Related Work

Active Learning. Many works have tackled the batch-mode active learning problem [Settles, 2009], where the goal is to optimally query an oracle for labels given a pool of unlabeled examples. In the streaming active learning setting, the entire pool is not available ahead of time. In both cases, many heuristics have been designed as acquisition functions, for eg. picking examples the model is most uncertain about, or that are closest to the margin, or based on density estimation. In our setting, we focus on the streaming active learning setting and compare how an active learning policy learned end-to-end via reinforcement learning compares with such heuristics.

Policy Active Learning. Recent work [Fang et al., 2017, Konyushkova et al., 2017, Bachman et al., 2017] have proposed learning policies for active learning from data, instead of using hand-designed heuristics. The primary intuition is that different heuristics may be optimal for different situations, and so learning data-driven acquisition functions might be the best solution. Additionally, many heuristics that rely on features such as model uncertainty underperform in the cold-start setting, as an untrained model tends to be uncalibrated. With policy active learning, the policy and model can co-train and this can help ameliorate the above issue. Finally, policy active learning is amenable to the *streaming* active learning setup, where we do not have access to all the data before hand [2], while most commonly employed heuristics are not. In our work, we focus on a similar setup as [Fang et al., 2017], but focus on the task of visual recognition (and design state representations conducive to images), and work with deep neural network as our model architecture (which presents challenges due to noisy rewards arising from local optimization). Further, we study *transfer* of learned policies across visual domains.

Transferring policies across domains. Some recent work has looked at transferring policies across domains. Combes et al. [2018] study data augmentation, meta-learning, and adversarial training approaches to learn transferable, task-agnostic policies. Fang et al. [2017] also look at transfer of active learning policies for NER to different languages, but make use of pretrained multilingual embeddings for the same. Similarly, Li et al. [2018] propose a meta-optimization strategy of simulating domain shift during training to learn invariant policies. Pang et al. [2018] propose a multitask training scheme to learn transferable active learning policies, and share a very similar motivation with this work – however, they assume access to multiple disjoint datasets, while we focus on low-resource transfer using only a single pass over the target domain data.

3 Approach

We follow the procedure described in Fang et al. [2017] in large part, with a few modifications. Given a data-rich source domain D, and an initially untrained model ϕ (parameterized as a convo-

[2] Note that it is equally applicable to the traditional pool-based setup.

lutional neural network) and we pose the active learning problem as a Markov Decision Process and learn a policy π as follows:

- **State.** At each timestep, the state of the MDP is characterized as a concatenation of a feature representation of the input datapoint x_i and the logits obtained from the model ϕ after classifying x_i. Specifically, we process the image using a series of convolutional and pooling layers with ReLU nonlinearities, and concatenate this representation with the logits obtained from the model π after a forward pass of the example x_i.
- **Action.** At each timestep, the policy either chooses whether to request a label for the current datapoint ($a_i = 1$) or not ($a_i = 0$) as its action. Note that if a label is requested, the model is trained with a few iterations of gradient descent on the new example [3].
- **Reward.** The policy π is given a reward $r(s_{i-1}, a)$ as the change in accuracy of the model ϕ after action a on a held-out validation set.
- **Budget.** The policy has a fixed budget B of the number of examples it can get labeled in a given episode. Once the budget is exhausted, the episode ends and the model is reinitialized with random weights.
- **Learning.** We experiment with deep Q-learning [Mnih et al., 2015] for learning the policy π, and run the learning algorithm for several episodes. Our policy is parameterized as a 3-layer deep neural network that takes in the state representation described above and maps is to a two-dimensional vector. The DQN is used using stochastic gradient descent with an objective of minimizing the mean square error between the Q-values predicted by the DQN and the expected values using the Bellman equation as $\mathcal{L}(\phi) = \mathbb{E}_{s,a,r,s'}\left[\left(r + \gamma \max_{a'} Q\left(s', a'; \phi_{i-1}\right) - Q(s, a; \phi)\right)^2\right]$. We store experiences (s_i, a, r, s_{i+1}) in an experience replay buffer and sample minibatches from this buffer while learning our policy. We also employ an ϵ-greedy strategy to encourage exploration.

Episodic training. The episodic training pipeline that we employ is outlined in Algo.1. Note that we shuffle the order of data between episodes.

Transfer. We study two modes of transfer. First, we study how well the learned policy directly transfers to a new domain, i.e. whether the policy implicitly learns domain invariant features. Second, we study explict transfer, by adding a simple "finetuning" strategy comprising of a single pass over our target domain that is used to update the policy.

Algorithm 1 Policy active learning for visual recognition, as proposed in Fang et al. [2017].

Input: Data D, Budget B, Replay Buffer M
Output: Active learning policy π ▷ Run this for a large number of episodes
for episode = $\{1, 2, ..N\}$ **do**
  Initialize ϕ randomly, shuffle D
  for $i \in \{1, 2, ..|D|\}$ **do**
    Construct state s_i from $\mathbf{x_i}$
    a_i = argmax $Q^{\pi}(s_i, a)$
    if a_i = 1 **then** ▷ Run few iterations of gradient descent on example
      Update model ϕ using (x_i, y_i)
    $r_i \leftarrow Acc(\phi_t) - Acc(\phi_{t-1})$ ▷ Estimate reward from change in validation accuracy.
    if $|D_l| = B$ **then**
      $s_{i+1} \leftarrow$ Terminate
    else
      Construct new state s_{i+1}
    Store (s_i, a_i, r_i, s_{i+1}) in replay buffer
    Sample random minibatch from replay buffer M
    Perform gradient descent step on policy π ▷ Train DQN

[3] We experimented with several strategies to reduce the variance in the reward cause due to noisy local optimization, details are discussed in Sec. 4

	$B = 10$	$B = 100$	$B = 1000$
Random	31.1 ± 9.2	64.3 ± 3.9	91.5 ± 0.2
Entropy	34.1	76.0	91.6
Margin	32.0	63.1	91.5
DQN (Ours)	33.5 ± 8.3	73.1 ± 1.3	92.1 ± 0.2

Table 1: Results for approach for varying budgets on MNIST.

4 Results

4.1 Experimental Setup

We present results on the task of digit recognition. Our digit recognition model is parameterized as a convolutional neural network with two conv-pool-relu blocks followed by two fully connected layers and a softmax layer. We study the streaming active learning setting, where we do not have access to future datapoints ahead of time and the model is retrained every time a new label is queried. We assume that our oracle always produces correct labels.

Datasets. We run experiments on MNIST [LeCun et al., 1998], and study transfer to USPS images. We create disjoint subsets of the training set of MNIST for the policy learning and policy evaluation stages, respectively.

Metrics. We employ the visual recognition accuracy on the test split of the target domain(s) for a given annotation budget as our metric.

Baselines. As baselines, we employ two traditional active learning acquisition functions – entropy sampling, and margin-based sampling. Entropy or uncertainty sampling picks the example that the model is most uncertain about, as measured by the entropy of its prediction scores, to be labeled. Margin-based sampling picks the example with the smallest separation between its top two predictions to be labeled [Wang and Shang, 2014]. Note that we employ both these approaches in a streaming setup (i.e. the model is retrained with each new label), and match the evaluation setup to the DQN evaluation setup exactly. However, these approaches are pool-based by design and access the entire pool of data, which makes them particularly strong baselines. Finally, we also include a random sampling baseline.

Hyperparameters. We use a learning rate of 3e-5 and a weight decay of 1e-4 with the Adam [Kingma and Ba, 2014] optimizer. We learn our policies for 1000 episodes, and shuffle the data order between episodes. We also pretrain our model with a small amount of data (100 train examples) to make sure that we do not start from completely uncalibrated models. We use a learning rate of 1e-4, a discounting factor γ of 0.99, and a τ of 1e-3 for the soft update of our DQN policy.

In practice, we found retraining the model with batches of 10 examples instead of a single example to be critical to make the policy converge. While this increases the difficulty of the credit assignment, we believe it helps significantly reduce the variance of the reward estimate arising from local optimization via gradient descent on a single example.

4.2 Policy Active Learning on MNIST

In this section, we present results for policy active learning on MNIST alone for varying budgets and compare that with baselines. Policies are learned using the algorithm described in Algo. 1. The policy learning and testing is done on disjoint subsets of the MNIST train set, and during learning rewards are obtained from the MNIST val set. Results are presented in Table 1. Performance is averaged over 3 runs, with the order being shuffled between runs. We report 95% confidence intervals. The learning employs an epsilon greedy strategy, and the policy is parameterized as a DQN with three fully connected layers and ReLU nonlinearities.

As seen in Table 1, we are able to learn good active learning policies on MNIST for a budget of 100, that considerably outperform the margin and random baselines but underperform against the entropy baseline. However, at a budget of 10, no statistically significant difference is observed from

	$B = 100$	$B = 1000$
Random	81.7 ± 0.8	95.2 ± 0.1
Entropy	83.3	95.1
Margin	84.9	94.6
DQN (Ours)	84.3 ± 1.4	95.8 ± 0.2
DQN-ft (Ours)	85.2 ± 2.7	96.0 ± 0.1

Table 2: Results for policy transfer from MNIST to USPS.

	Accuracy
Logits only	68.3 ± 3.7
Pixels only	67.9 ± 1.6
Logits + Pixels (Conv)	73.1 ± 1.3

Table 3: Results for different ablations for our proposed model for $B = 100$.

a random policy, most likely due to high-variance in the rewards [4]. At $B = 1000$, the DQN policy outperforms all baselines. However, diminishing returns are observed from active learning in itself with the relatively large amount of the data, with the difference observed being rather small.

As a sanity check, we tried to visualize the examples picked and rejected by the policy with the highest confidence, to interpret trends. However, the qualitative results were difficult to interpret and we could not see clear trends of what the policy was optimizing for.

4.3 Policy transfer to USPS

Table 2 describes the results of transferring our policies learned on MNIST to USPS, for different budgets. The policy finetuning happens on the USPS train set, and rewards are obtained from the USPS val set. Final performance is reported on the USPS test set.

We find that for a budget of 100, our approach outperforms the random and entropy baseline and is on par with the margin based baselines. Further, finetuning leads to a 1% improvement in mean accuracy. However, we note that even a random baseline performs quite well and gains are modest, which potentially points to active learning having limited usefulness in this domain. With a budget of 1000, we observe similar trends, but the differences are much narrower and likely not statistically significant.

4.4 Model ablations

In Table 3 we report performance for different ablations of our model that we obtain by varying our state representation, on the MNIST test set using a budget of 100. Logits only and pixels only refers to only using flattened vectors of logits and pixels respectively as the state representation. Finally, our best performing approach is Logits + Pixels (Conv) which uses a convolutional feature representation of the image concatenated with the prediction logits from the model.

5 Conclusion and Future Work

In this project, we demonstrated how we could learn effective policies for active learning for the task of visual recognition using deep reinforcement learning. We further studied the transferability of such learned policies to a low-resource domain, but we did not see promising results on transfer. Studying ways to explicitly incorporate domain invariance into the policy learning and finetuning stages would be a natural next step towards resolving this issue.

Many interesting extensions are possible, especially for transfer, that we were unable to experiment with due to shortage of time. Unsupervised adversarial feature-level adaptation can be employed as

[4] I also did not spend a lot of time tuning hyperparameters for the $B = 10$ setting due to lack of time.

done in Tzeng et al. [2017] and then a policy can be learned on this aligned space, which would be transferable to the target domain by design. Further, also adding pixel-level adaptation as done in Hoffman et al. [2017] could lead to further improvements. Finally, experimenting with different policy parameterizations, including double DQN's, dueling DQN's prioritized experience replay, as well as using policy gradients etc. are interesting alternatives that are worth experimenting with.

References

Philip Bachman, Alessandro Sordoni, and Adam Trischler. Learning algorithms for active learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 301–310. JMLR. org, 2017.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

Remi Tachet des Combes, Philip Bachman, and Harm van Seijen. Learning invariances for policy generalization. *arXiv preprint arXiv:1809.02591*, 2018.

Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. *arXiv preprint arXiv:1708.02383*, 2017.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Learning active learning from data. In *Advances in Neural Information Processing Systems*, pages 4225–4235, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Kunkun Pang, Mingzhi Dong, Yang Wu, and Timothy Hospedales. Meta-learning transferable active learning policies by deep reinforcement learning. *arXiv preprint arXiv:1806.04798*, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

Dan Wang and Yi Shang. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE, 2014.

Xiangxin Zhu, Carl Vondrick, Deva Ramanan, and Charless C Fowlkes. Do we need more training data or better models for object detection?. In *BMVC*, volume 3, page 5. Citeseer, 2012.