

Episode-Based Active Learning with Bayesian Neural Networks

Feras Dayoub, Niko Sünderhauf, Peter Corke
Australian Centre for Robotic Vision
Queensland University of Technology (QUT)
Brisbane, 4000 QLD, Australia

{feras.dayoub, niko.suenderhauf, peter.corke}@qut.edu.au

Abstract

We investigate different strategies for active learning with Bayesian deep neural networks. We focus our analysis on scenarios where new, unlabeled data is obtained episodically, such as commonly encountered in mobile robotics applications. An evaluation of different strategies for acquisition, updating, and final training on the CIFAR-10 dataset shows that incremental network updates with final training on the accumulated acquisition set are essential for best performance, while limiting the amount of required human labeling labor.

1. Introduction

Obtaining labeled training data is a big challenge for a robotic scene understanding system that is pre-trained on a dataset, but then has to adopt to a real world deployment environment. Active learning, [Cohn et al.(1996)Cohn, Ghahramani, and Jordan], helps to minimize the necessary human labeling labor by acquiring only the most informative samples from a pool of available images.

In the context of mobile robotics, active learning happens *episodically*. Starting with an initial classifier, the robot encounters a stream of images while performing its mission. After a certain time (an *episode*), an acquisition function determines the most informative encountered images, and a human can be asked to provide ground truth labels. The initial network is then updated using this set of acquired images. The process repeats as new, previously unseen images are encountered during the next episode.

In this paper we investigate different strategies of performing active learning in this episode-based scenario. We aid the acquisition function by using Bayesian deep networks as classifiers. Our goal is to enable a mobile robot to adopt its perception system to its deployment environment with as little human help and interaction as possible.

2. Episode-based Active Learning

Notation We use \mathbf{N}_t to represent a network obtained after episode t and write $\mathbf{N}_t = \mathbf{N}_{t-1} \otimes \{\mathcal{A}\}$ to express fine-tuning network \mathbf{N}_{t-1} with the data set $\{\mathcal{A}\}$ to obtain \mathbf{N}_t .

Problem Definition We define the problem of episode-based active learning as follows: Start with an initial, pre-trained network \mathbf{N}_0 . For episodes $t = 1 \dots k$ perform the following steps: (1) present an episode of n previously unseen and unknown images to the network \mathbf{N}_{t-1} , and obtain classification results $p(\mathbf{x}_i)$, $i = 1 \dots n$; (2) based on $p(\mathbf{x}_i)$, use an acquisition function to determine the set \mathcal{A}_t of most informative images and ask an oracle (a human) for their ground truth labels; (3) update the network \mathbf{N}_{t-1} by fine tuning with the acquired dataset \mathcal{A}_t . After stopping the active learning process after k episodes (e.g. based on the number of acquired images per episode, or simply based on the number of passed episodes k), a final training step might be performed to obtain the final network \mathbf{N}_f .

Different strategies are available for updating the network, for the final training, and for the acquisition function. Table 1 lists the different combinations of strategies we investigated and evaluated in this paper.

Network Update Updating the network after every episode can be done using only the newly acquired set \mathcal{A}_t from that episode, or by using the full data $\cup_{i=1}^t \mathcal{A}_i$ that has been acquired so far. We will evaluate four different methods for updating the network \mathbf{N}_{t-1} : incremental fine-tuning using only the most recently acquired images: $\mathbf{N}_t = \mathbf{N}_{t-1} \otimes \{\mathcal{A}_t\}$; incremental updating using the growing set of all acquired images so far: $\mathbf{N}_t = \mathbf{N}_{t-1} \otimes \{\cup_{i=1}^t \mathcal{A}_i\}$; and fine-tuning on the initial network: $\mathbf{N}_t = \mathbf{N}_0 \otimes \{\cup_{i=1}^t \mathcal{A}_i\}$, and $\mathbf{N}_t = \mathbf{N}_0 \otimes \{\mathcal{A}_t\}$.

Acquisition Function We apply a maximum entropy acquisition strategy that selects the images with the highest classification uncertainty from an episode.

[Gal et al.(2017)Gal, Islam, and Ghahramani] showed this strategy to be competitive to more complex acquisition schemes. For every image \mathbf{x}_i in the current episode, we obtain the class probability distribution $p(\mathbf{x}_i)$ from the current network \mathbf{N}_t . An image is acquired if $H(p(\mathbf{x}_i)) > \theta$, where $H(\cdot)$ denotes the entropy and θ is a threshold parameter.

In our evaluation, the true class probability distribution $p(\mathbf{x}_i)$ is approximated by a Bayesian neural network approach proposed by [Gal & Ghahramani(2015)Gal and Ghahramani]. We enable Dropout during test time, pass every image through the network 64 times, and average over the obtained 64 distributions.

Final Training The final network after k episodes can be obtained either by simply using the network \mathbf{N}_k from the last update step, or by using all acquired images to perform an additional fine-tuning: $\mathbf{N}_f = \mathbf{N}_k \otimes \{\cup_{i=1}^k \mathcal{A}_i\}$.

3. Evaluation on CIFAR-10

Dataset We evaluate the 6 different strategies in Table 1 using the CIFAR-10 dataset. This standard dataset consists of 32×32 pixel RGB images of ten classes and is split into 50,000 images for training and 10,000 for testing. We divide the training set into 10 splits of 5k images each. The first split is used to train the initial model \mathbf{N}_0 that is shared between strategies. The rest of the non-overlapping splits is used as 9 episodes. All the testing is done on the 10k test images.

Network Architecture We use a simple network consisting of 4 convolution layers (2×32 and 2×64 channels, convolutions are all 3×3) followed by one fully connected layer (512 units) and a 10-fold Softmax layer. Dropout ($p = 0.5$) is used after every layer, and max-pooling layers after conv layers 2 and 4. The network is trained with Adam, [Kingma & Ba(2015)Kingma and Ba], and early stopping.

Evaluation Protocol For every strategy in Table 1 we evaluate the test set accuracy of the final network that would result from stopping the active learning after every episode. All experiments are performed 10 times and the average accuracy and acquired number of training images are reported. To compare the performance of different strategies, we furthermore define an efficiency score as $\xi = \frac{\text{test accuracy}}{\text{fraction of used training set}}$. The efficiency of a network is high when it uses less training samples to gain a higher accuracy on the test set. Furthermore the *relative efficiency* is defined as $\frac{\xi}{\xi_F}$, where ξ_F is the efficiency of the network trained on the full training set. We empirically set the acquisition threshold to $\theta = 0.8$, but also analysed the influence of this parameter.

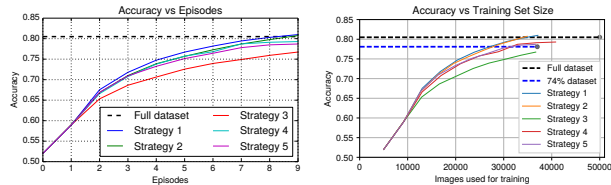


Figure 1. Test set accuracy for various active learning strategies. Incremental fine-tuning with final training on the accumulated acquired images (strategies 1 and 2) outperforms other variants and even reaches slightly better performance than the baseline model that was trained on the full dataset. Plots averaged over 10 trials.

Results Our evaluation showed that strategies 1 and 2 which use incremental fine-tuning during the update step with a final training step on the accumulated acquired images outperform the other strategies. In particular, strategies 4 and 5 that update based on the initial network \mathbf{N}_0 perform worse. This is illustrated in the plots of Fig. 1, where strategies 1 and 2 learn faster (i.e. gain more accuracy per episode) and more efficient (i.e. gain more accuracy per acquired images). Both strategies even outperform the baseline model (training on the full dataset) by a small margin. Strategy 2 is more data-efficient, using 4% less training data while only sacrificing 0.2% of test accuracy. This comes at an increased computational cost for the update step, which is in $\mathcal{O}(n)$ instead of $\mathcal{O}(1)$ due to the growing update fine-tuning dataset $\cup_{i=1}^t \mathcal{A}_i$. Training on randomly selected 74% of the training data results in a significantly worse performance, which underlines the efficacy of the proposed active learning strategies (dashed blue line in Fig. 1 (right)).

The differences between strategy 3 and the first two strategies illustrate the importance of using the accumulated acquired images $\cup_{i=1}^k \mathcal{A}_i$ during updating or at least during final training. We assume this prevents the *forgetting* that seems to occur when using only the most recent acquired images.

Fig. 2(left) compares the number of images acquired by every strategy over the 9 episodes. Strategy 2 is very data-efficient, acquiring the least amount of images, while reaching the second highest accuracy. This is expressed in its relative efficiency score, that is the highest of all strategies. Notice how strategy 4 always acquires roughly the same amount of images. This is not surprising given its update strategy uses the initial network \mathbf{N}_0 and only the most recent acquisition set \mathcal{A}_t , thereby not learning more about the environment over time as the other strategies do.

Fig. 2(right) reveals the influence of the parameter θ used as a threshold in the maximum entropy acquisition function. As expected, a higher threshold acquires fewer images, which ultimately leads to a smaller training set and sacrifices accuracy.

Strat.	Network Update	Final Training	Accur.	Used Training Set	Relative Efficiency
1	$\mathbf{N}_t = \mathbf{N}_{t-1} \otimes \{\mathcal{A}_t\}$	$\mathbf{N}_f = \mathbf{N}_k \otimes \{\cup_{i=1}^k \mathcal{A}_i\}$	0.810	74%	1.36
2	$\mathbf{N}_t = \mathbf{N}_{t-1} \otimes \{\cup_{i=1}^t \mathcal{A}_i\}$	$\mathbf{N}_f = \mathbf{N}_k$	0.808	70%	1.43
3	$\mathbf{N}_t = \mathbf{N}_{t-1} \otimes \{\mathcal{A}_t\}$	$\mathbf{N}_f = \mathbf{N}_k$	0.767	73%	1.31
4	$\mathbf{N}_t = \mathbf{N}_0 \otimes \{\mathcal{A}_t\}$	$\mathbf{N}_f = \mathbf{N}_0 \otimes \{\cup_{i=1}^k \mathcal{A}_i\}$	0.793	82%	1.20
5	$\mathbf{N}_t = \mathbf{N}_0 \otimes \{\cup_{i=1}^t \mathcal{A}_i\}$	$\mathbf{N}_f = \mathbf{N}_0 \otimes \{\cup_{i=1}^k \mathcal{A}_i\}$	0.787	73%	1.34
6	regular training on the full training set		0.805	100%	1.0
7	regular training on random 74% of the full training set		0.781	74%	1.31

Table 1. Different active learning strategies evaluated in this paper. Accuracy on the test set, the size of the used training set and the relative efficiency are reported for the final network N_f after all 9 episodes.

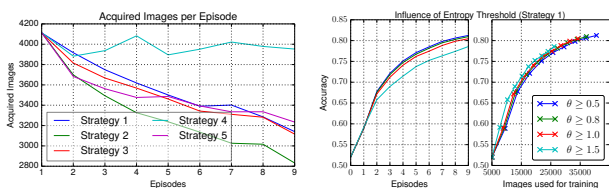


Figure 2. (left) The acquisition function and network update strategy determine the data-efficiency: Strategy 2 acquires the least number of images, but outperforms even the baseline model in terms of test set accuracy. (right) Influence of the selection threshold θ in the max entropy acquisition function evaluated on strategy 1.

4. Conclusions

Our evaluation found incremental update strategies with a final training step based on the accumulated acquired image set performed best for episode-based active learning with Bayesian neural networks. In contrast to earlier work such as [Islam(2016), Gal et al.(2016)Gal, Islam, and Ghahramani] that demonstrated active learning ideas with Bayesian networks in a pool-based setup on MNIST, we presented experiments on the more challenging CIFAR-10 dataset. Furthermore, we evaluated on an *episode*-based scenario where the system does not get a chance to re-observe images it did not choose to acquire. This scenario is closer to the requirements encountered by a robot in reality.

Acknowledgments

This research was conducted by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016).

References

[Cohn et al.(1996)Cohn, Ghahramani, and Jordan]
David A Cohn, Zoubin Ghahramani, and Michael I

Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4(1):129–145, 1996.

[Gal & Ghahramani(2015)Gal and Ghahramani] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

[Gal et al.(2016)Gal, Islam, and Ghahramani] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Active Learning with Image Data. In *Workshop on Bayesian Deep Learning, NIPS*, 2016.

[Gal et al.(2017)Gal, Islam, and Ghahramani] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*, 2017.

[Islam(2016)] Riashat Islam. Active Learning for High Dimensional Inputs using Bayesian Convolutional Neural Networks. Master’s thesis, University of Cambridge, 2016.

[Kingma & Ba(2015)Kingma and Ba] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proc. of International Conference on Learning Representations (ICLR)*, 2015.