DEEP ADVERSARIAL ACTIVE LEARNING WITH MODEL UNCERTAINTY FOR IMAGE CLASSIFICATION

Zheng Zhu, Hongxing Wang⋆

Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University)
Ministry of Education, China
School of Big Data & Software Engineering
Chongqing University, Chongqing, China

ABSTRACT

Active learning aims at selecting and labeling as few samples as possible to train a good task model. Most existing methods rely on various heuristics to iteratively select a single sample in each active learning loop, thus cannot tackle large datasets efficiently. In this paper, we propose a new batch-mode active learning method, which can plug model prediction uncertainty into adversarial batch selection to ensure the selected samples are representative in unlabeled data, complementary to labeled data, and beneficial for model training. Experiments on four benchmark image datasets validate the effectiveness and efficiency of the proposed method for active image classification in comparison with the state-of-the-art methods.

Index Terms— Active learning, Adversarial learning, Uncertainty, Image classification

1. INTRODUCTION

In the past few years, the rapid development of deep learning has made remarkable successes across multiple vision applications [1, 2]. However, training a deep neural network usually needs a large amount of labeled data [3], which is quite time-consuming and laborious. Thus, there has been increased attention in deep learning with limited labeled data. It has been of importance to determine which data to be labeled, being in line with the goal of active learning (AL) [4].

Active learning has been studied for decades. Many heuristic methods, like uncertainty sampling [4, 5], query-by-committee [6], and expected model change [7], have been shown to be effective in traditional machine learning on small datasets. However, most of these methods select training samples one by one, which cannot efficiently bring large training data for deep learning. Therefore, it is of great interest for deep learning to query a batch of samples for active labeling in each active learning loop, which is known as batch-mode active learning.

⋆Corresponding author.

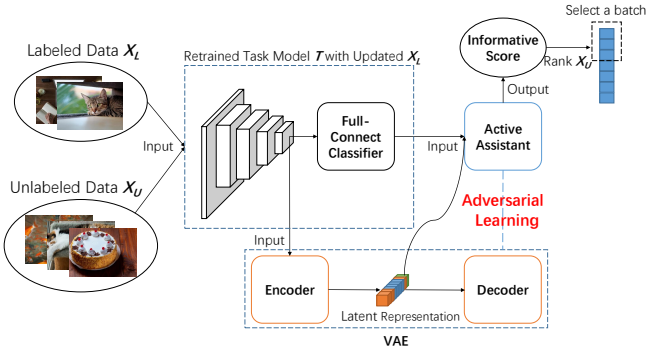


Fig. 1. Overview of the proposed AALU in one active learning loop.

For batch-mode active learning, some methods focus on leveraging generative adversarial models such as GAN [8], conditional GAN [9] and VAE-ACGAN [10] to generate batch samples for deep active learning [11, 12, 10]. However, these GAN-based methods will also bring additional training cost on generative samples. To avoid training extra samples, recent efforts have been made towards adversarial discrimination between labeled and unlabeled data for distinct selection of batch samples [13, 14].

Despite the above advances on adversarial selection of batch samples, the prediction uncertainty of actively learned model is unfortunately not considered. As a result, the selected samples may be of little help to improve the model prediction performance. To address this issue, we propose a new Deep Adversarial Active Learning with Model Uncertainty (AALU). Fig. 1 illustrates one active learning loop of AALU. We pass both labeled data X_L and unlabeled data X_U through the task model T, which has been trained with updated X_L, followed by obtaining the features and predictions of input data. The features are further fed into a Variational Auto-Encoder (VAE) [15] to acquire corresponding latent representations. Then, we concatenate the predictions with the latent representations and input them to a trained active assistant to output the informative scores of X_U. The VAE and the ac-

tive assistant are trained in an adversarial manner. Finally, we rank the unlabeled samples with their informative scores and select a batch of samples for labeling. Our proposed AALU has the following advantages:

- As we introduce model uncertainty into deep adversarial active learning, the proposed AALU can select those uncertain samples for follow-up labeling and deep model learning to guarantee the prediction performance.
- Due to the combination of model prediction uncertainty and adversarial batch selection, the selected samples by the proposed AALU can reach a balance among being representative in unlabeled data, being complementary to labeled data, and being beneficial for model training.

For evaluation, we conduct image classification experiments on four benchmark image datasets, which show the effectiveness and efficiency of the proposed AALU in comparison with the state-of-the-art methods.

2. PROPOSED METHOD

In the initial setting of our proposed AALU, there exist a large pool of unlabeled image data, X_U, and a small pool of labeled image data, X_L. We denote by (x_L, y_L) a sample x_L with label y_L in X_L, and denote by x_U a sample in X_U. In each active learning loop, a task model T is trained on existing X_L, and an active assistant A is trained based on the current T, x_L, and x_U to choose b samples moving from X_U into X_L. After that, the selected b samples are labeled by an external oracle. This process is repeated until the labeling budget B, i.e., the number of samples allowed to be labeled, is exhausted.

2.1. Variational adversarial representation learning

To avoid choosing samples similar to the labeled, it is vital important to train an active assistant A to measure the difference between the pools of labeled and unlabeled data. This requires mapping both labeled and unlabeled data into a common representation space. VAE has been demonstrated to be highly effective in representation learning, based on which, a variational adversarial active learning has also been proposed in [14]. In this paper, we also use the adversarial VAE to learn a low dimensional representation space that can identify the input data distribution. Instead of taking raw images as the input of VAE, we input VAE with the features extracted from raw images using the trained task model. By this way, not only is our VAE easier to be trained, but it also encodes task-related semantic information, which will benefit sample selection for the target task. The VAE in our AALU is to minimize the variational lower bound on the marginal likelihood of image feature f:

$$\mathcal{L}_{vae}^{rep} = \mathbb{E}[\log(E(f|z))] - \mathrm{KL}(D(z|f)\|p(z)), \tag{1}$$

where E and D are the encoder and decoder, respectively; z is the latent vector encoded from f; KL is the Kullback-Leibler divergence function [16]; and $p(z)$ is a probability density function that follows standard Gaussian distribution. Note that minimizing Eq. (1) does not need any label information, thus both X_U and X_L could be used. Meanwhile, we cannot directly use Eq. (1) to distinguish between labeled and unlabeled images for active learning. Therefore, there needs an active assistant A to incorporate the data source information. Similar to [14], we train VAE and A in an adversarial manner. On the one hand, VAE intends to make A indistinguishable between z_L (z for label data) and z_U (z for unlabeled data), i.e., let $A(z_L) = A(z_U)$ hold. The adversarial loss for VAE is thus defined by cross-entropy in the following:

$$\mathcal{L}_{vae}^{adv} = -\mathbb{E}[\log(1 - A(z_L)] - \mathbb{E}[\log(1 - A(z_U)]. \tag{2}$$

Combining Eq. (1) and Eq. (2), we can obtain the full objective loss of the adversarial VAE as:

$$\mathcal{L}_{vae} = \mathcal{L}_{vae}^{rep} + \beta\mathcal{L}_{vae}^{adv}, \tag{3}$$

where β is a hyperparameter to balance between $\mathcal{L}_{vae}^{rep}$ and $\mathcal{L}_{vae}^{adv}$. On the other hand, A intends to discriminate if the sample is from labeled data pool or not. It means that A is trained to correctly assign z_L to labeled class ($A(z_L) = 0$) and z_U to unlabeled class ($A(z_U) = 1$). Thus the adversarial loss for A is defined as:

$$\mathcal{L}_{A}^{adv} = -\mathbb{E}[\log(1 - A(z_L))] - \mathbb{E}[\log(A(z_U))]. \tag{4}$$

After co-training with VAE, A will output higher scores for those unlabeled samples that are most different from the labeled data. Therefore, the outputs of A on unlabeled data can serve as informativeness measures for active selection.

2.2. Learning with model uncertainty

Through the adversarial training between VAE and A, we can identify the difference between labeled and unlabeled distribution and select samples by A for active learning. However, the selection is almost independent of model predictions on the target task, which may not bring a satisfactory result. To obtain a predication dependent selection, we add uncertainty information of the task model to A. For simplicity, we take the image classification task as an intuitive instance to explain the rationale behind our uncertainty module. For this task, we work on learning a latent representation z and a classification prediction p for each input image. Actually, p indicates the model uncertainty when classifying the corresponding image. We then concatenate z and p for a new representation for each image, which is denoted as m. As a result, we revise the input of A from z to m, which implies the following revisions of adversarial loss for VAE in Eq. (2), the total loss for adversarial

VAE in Eq. (3), and the adversarial loss for A in Eq. (4):

$$\hat{\mathcal{L}}_{vae}^{adv} = (-\mathbb{E}[\log(1 - A(m_L)] - \mathbb{E}[\log(1 - A(m_U)]) \quad (5)$$
$$\hat{\mathcal{L}}_{vae} = \mathcal{L}_{vae}^{rep} + \beta\hat{\mathcal{L}}_{vae}^{adv} \quad (6)$$
$$\hat{\mathcal{L}}_{A}^{adv} = -\mathbb{E}[\log(1 - A(m_L))] - \mathbb{E}[\log(A(m_U))]. \quad (7)$$

As samples with bigger prediction losses are more uncertain for task model training, we also impose the ranked loss in [17] on A to enable A to measure the model uncertainty after training:

$$\mathcal{L}_{A}^{loss} = \frac{1}{n}\sum_{i=1}^{n} \max(0, -\text{sgn}(l_i - l_{n+i})(A(m_i) - A(m_{n+i})) + \xi), \quad (8)$$

where l_i is the prediction loss of the task model for the i^{th} labeled sample in a mini-batch with $2n$ samples; sgn$(*)$ is a sign function which returns the sign of given input; and ξ is the pre-defined positive margin for $A(m_i)$ and $A(m_{n+i})$. By combining the revised adversarial loss in Eq. (7) and the ranked loss in Eq. (8), we obtain the total loss for A:

$$\mathcal{L}_A = \hat{\mathcal{L}}_A^{adv} + \lambda\mathcal{L}_A^{loss}, \quad (9)$$

where λ is a hyperparameter to balance $\hat{\mathcal{L}}_A^{adv}$ and $\mathcal{L}_A^{loss}$. According to Eq. (9), we can train the active assistant to rank and select those representative samples that are most complementary to existing labeled data in terms of the task model.

2.3. Model training and active sampling

To perform active learning, the active assistant A and the VAE in Eq. (6) and Eq. (9) are jointly co-trained. After training A, we can directly use it to select the most informative samples. Those b samples with bigger scores of A will be sent to the oracle for labeling. Then all modules are updated using the augmented labeled dataset and reduced unlabeled pool. We show one active learning loop of the proposed AALU in Alg. 1.

Algorithm 1 AALU in One Active Learning Loop.

Input: Initial labeled pool (X_L, Y_L); Unlabeled pool X_U; Initialized parameters in task model T, VAE, Active assistant A; Number of epochs e.

for $epoch = 1...e$ **do**
  for all batches $(x_L, y_L) \sim (X_L, Y_L)$ **do**
    Minimize the task model loss to update T
  end for
end for
return The well trained T
Acquire prediction p_L for X_L, and p_U for X_U using T
for $epoch = 1...e$ **do**
  for all batches $(x_L, y_L) \sim (X_L, Y_L)$ and $x_U \sim X_U$ **do**
    Minimize loss $\hat{\mathcal{L}}_{vae}$ in Eq. (6) to update VAE
    Minimize loss $\hat{\mathcal{L}}_A$ in Eq. (9) to update A
  end for
end for
return The well trained VAE and A

3. EXPERIMENTAL EVALUATION

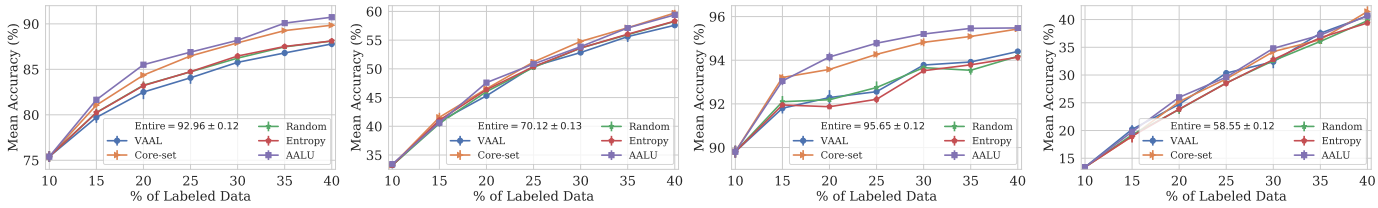
We evaluate the proposed AALU in image classification. Four different public datasets are used for the experiment, which are CIFAR10\100 [18], SVHN [19] and Caltech-256 [20]. CIFAR10 and CIFAR100 are composed of images in 10 classes and 100 classes, respectively, both including 50k training samples and 10k test samples. SVHN is a real-world digit dataset, from 10 classes of house numbers in Google Street View images, which has 73257 images for training and 26032 images for test. Caltech-256 is a dataset of 30607 object images in 256 categories, for which we randomly choose 2560 images (10 images per class) as the test set and let the remaining as the training set. We perform data augmentation for each dataset using random horizontal flips.

We compare our method with the following batch-mode active learning methods:

- **Random Selection**: randomly choosing samples for label query.
- **Entropy**: choosing samples with top entropy values for label query [21].
- **Core-set**: choosing samples most far from the labeled dataset by the K-Center-Greedy algorithm in [22].
- **VAAL**: choosing samples by the variational adversarial active learning proposed in [14].

We have implemented all the compared methods using PyTorch [23]. For each dataset, we start active learning with 10% of labeled data by randomly sampling from the entire dataset. In each active learning loop, we select 5% of unlabeled data for labeling and train a new classification model with updated labeled data. The classification model used in each experiment is ResNet18 [24]. In our method, we always set β to 1, ξ to 0.1, and λ to 10. We take the feature maps before full-connected layer as the extracted features for each input image. We use 3-layer multilayer perceptron (MLP) to build the encoder and decoder in VAE, as well as the active assistant. Adam [25] is used as the optimizer with the same learning rate of 5×10^{-4} and batch size of 128. We train each module in our method 100 epochs. For a fair comparison, we also train the VAE and discriminator in VAAL[14] 100 epochs. In each comparison, we randomly run each involved method five times and use the mean classification accuracy for evaluation.

The active learning results on different datasets are reported in Fig. 2. On CIFAR10, compared with the 92.96% accuracy obtained using the entire training data, our proposed AALU can reach 90.73% accuracy by only selecting 40% of the training data. Since our method can not only learn to identify the difference between labeled and unlabeled data distribution, but also can give priority to the data points that have



(a) CIFAR10 (b) CIFAR100 (c) SVHN (d) Caltech-256

Fig. 2. Performance of the proposed AALU in classification on CIFAR10, CIFAR100, SVHN, and Caltech-256 compared with VAAL, Core-set, Entropy, and Random Sampling.

Table 1. Ablation results with different proportions of labeled data (%) on CIFAR10.

Method	15	20	25	30	35	40
Random	80.25	83.25	84.72	86.23	87.47	88.11
advVAE	80.75	81.90	85.02	86.88	86.55	88.43
advVAE+Prediction	81.55	84.44	**87.54**	88.13	88.50	89.64
lossLearn	80.17	83.33	85.17	85.22	87.03	87.21
advVAE+lossLearn	81.32	84.23	86.91	**88.65**	88.94	90.18
AALU	**81.65**	**85.51**	86.89	88.19	**90.08**	**90.73**

the greatest impact on the classification model, it can achieve better results than others on each dataset. On SVHN, the maximum achievable mean accuracy is 95.65% using 100% of the training data while AALU can reach 95.2% with 30% of training data. It further validates the effectiveness of AALU in informative sample selection. From Fig. 2, we can observe that our method is less effective on CIFAR100 and Caltech-256 than on CIFAR10 and SVHN, but still achieves good results comparable to state-of-the-art methods. This may be caused by the difficulty in extracting information from the predicted results on image data with more categories. It is worth nothing that the results of VAAL are different from those reported in [14] due to different epoch settings. Specifically, we train each module of VAAL in 100 epochs for a fair comparison.

We conduct an ablation study to demonstrate the effectiveness of different design choices in AALU. As show in Table 1, the variants of ablations we considered include:

- **advVAE**: training VAE with an active assistant in adversarial manner using the loss in Eq. (3).
- **advVAE+Prediction**: based on advVAE, but adding model prediction results to the input of the active assistant.
- **Losslearn**: training an active assistant only using the loss in Eq. (8).
- **advVAE+Losslearn**: based on advVAE and lossLearn, but without prediction input.

The results show that the proposed prediction module can improve the active learning performance compared with the adversarial VAE without prediction as input. After adding a loss item in Eq. (8) to the loss function of A, the performance of adversarial VAE can be significantly improved. The ablation results at different selection ratios of training data show the effectiveness of our design strategy in AALU.

Table 2. Active query time (s) per 100 samples with different methods on CIFAR10.

Core-Set [22]	VAAL [14]	AALU
29.01	75.70	**14.07**

We also conduct active query time comparison on CIFAR10 using a single NVIDIA TITAN V. Experimental settings are the same for the compared methods. The active query time includes the time for algorithm training and sample selection. The results in Table 2 show that our method takes only 14.07s on average to query 100 samples, which is much faster than others. This should be mainly attributed to the compact network structure of our method.

4. CONCLUSION

In this paper, we introduce an adversarial active learning method that takes the model uncertainty into account. By aid of representation of an active assistant, the proposed method can embed model prediction uncertainty into adversarial sample selection, which improves the effectiveness of data labeling in image classification. Moreover, the proposed method can take less query time in one active learning loop compared with the state-of-the-art methods, thus being more practical. In the future, we would like to extend the proposed method for other vision-related tasks besides image classification.

5. ACKNOWLEDGEMENTS

This work is supported in part by the National Natural Science Foundation of China under Grants 61976029 and 61602069, the Chongqing Research Program of Basic Research and Frontier Technology under Grant cstc2016jcyjA0468, and the Fundamental Research Funds for the Central Universities under Grant 2018CDXYRJ0030.

6. REFERENCES

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, Nature Publishing Group, 2016.

[3] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[4] Burr Settles, "Active learning literature survey," Tech. Rep., University of Wisconsin-Madison, 2009.

[5] Simon Tong and Daphne Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, no. Nov, pp. 45–66, 2001.

[6] Ran Gilad-Bachrach, Amir Navot, and Naftali Tishby, "Query by committee made real," in *Advances in Neural Information Processing Systems*, 2006.

[7] Raphael Sznitman and Bruno Jedynak, "Active testing for face detection and localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1914–1920, 2010.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014.

[9] Mehdi Mirza and Simon Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[10] Toan Tran, Thanh-Toan Do, Ian Reid, and Gustavo Carneiro, "Bayesian generative active deep learning," in *International Conference on Machine Learning*, 2019.

[11] Melanie Ducoffe and Frederic Precioso, "Adversarial active learning for deep networks: a margin based approach," *arXiv preprint arXiv:1802.09841*, 2018.

[12] Dwarikanath Mahapatra, Behzad Bozorgtabar, Jean-Philippe Thiran, and Mauricio Reyes, "Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018.

[13] Yue Deng, KaWai Chen, Yilin Shen, and Hongxia Jin, "Adversarial active learning for sequences labeling and generation.," in *International Joint Conferences on Artificial Intelligence*, 2018.

[14] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell, "Variational adversarial active learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[15] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, 2014.

[16] Solomon Kullback, *Information theory and statistics*, Courier Corporation, 1997.

[17] Donggeun Yoo and In So Kweon, "Learning loss for active learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[18] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," Tech. Rep., University of Toronto, 2009.

[19] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[20] Gregory Griffin, Alex Holub, and Pietro Perona, "Caltech-256 object category dataset," Tech. Rep., California Institute of Technology, 2007.

[21] Burr Settles and Mark Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008.

[22] Ozan Sener and Silvio Savarese, "Active learning for convolutional neural networks: A core-set approach," in *International Conference on Learning Representations*, 2018.

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.

[24] 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*, 2016.

[25] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *International Conference for Learning Representations*, 2015.