Adversarial Active Learning for Deep Networks: a Margin-Based Approach

AAAI Press
Association for the Advancement of Artificial Intelligence
2275 East Bayshore Road, Suite 160
Palo Alto, California 94303

Abstract

We propose a new active learning strategy designed for deep neural networks. The goal is to minimize the number of data annotations queried by an oracle during training. Previous active learning strategies scalable for deep networks are mostly uncertainty-based. Margin-based approaches which focus on samples lying close to the decision boundary have been demonstrated to be among the most efficient techniques for shallow classifiers. However, the calculation of the margin for deep networks is intractable. In this article, we introduce a new active learning strategy: building on the margin-based learning theory, we leverage adversarial attacks to approximate the distance to the decision boundary as the smallest distance between unlabeled samples and their adversaries. Because an example and its adversary get, by definition, the same label, we get twice as much labeled data for the same number of provided annotations. Furthermore, our method benefits from the transferability of adversarial examples: we can build actively the training set for a first network and transfer it as it is to a second one. This approach outperforms random selection for training this second network. We compare to most recent deep active learning strategies like CORE-SET and demonstrate empirically that adversarial active queries yield faster convergence of CNNs trained on MNIST, the Shoe-Bag, Quick-Draw, and CIFAR datasets, requiring fewer annotations. For instance, our active learning strategy DFAL queries up to 33% fewer samples than CORE-SET, the current state-of-the-art, on MNIST and Quick-Draw for both LeNet5 and VGG8 networks. This gap is even more significant with other techniques (up to 80% for CEAL). Our method is also faster than CORE-SET with gains of up to 16% in running-time.

Introduction

The efficiency of deep networks is well known on large datasets. However, gathering and annotating massive dataset for supervised learning may prohibit the expansion of deep networks towards new fields such as chemistry or medicine (Smith et al. 2018; Hoi et al. 2006). A possible solution to obtain an efficient but reduced training set is to rely on active learning. Active learning is a family of methods seeking to select the smallest training dataset and only progressively append it as needed for training convergence, striving to limit the total number of required annotations. It is only recently that (Gal, Islam, and Ghahramani 2016; Zhou, Chen, and Wang 2010; Lin and Parikh 2017; Asghar et al. 2017; Zhu and Bento 2017), among others, have scaled active learning heuristics on deep networks, especially CNNs. However, transposing directly existing active learning on deep networks is not intuitive. First, scaling them to a high dimensional parameter networks may turn out to be intractable: some classic active learning methods such as Optimal Experimental Design (Yu, Bi, and Tresp 2006) require to inverse the Hessian matrix of the models at each iteration, which would be intractable for most CNNs. Secondly, one of the most standard active learning strategies is to rely on the uncertainty. We evaluate the uncertainty in deep networks through the network's output. However, this is known to be misleading. Indeed, the discovery of adversarial examples has demonstrated that measuring the uncertainty given the prediction may be overconfident. Adversarial examples are inputs modified with small (sometimes not perceptually distinguishable) but specific perturbations which result in an unexpected misclassification despite the strong confidence of the network in the predicted class (Szegedy et al. 2014). On the one hand, the existence of such adversarial examples leading to a wrong prediction with very high confidence brings into question uncertainty-based strategies. On the other hand, the magnitude of adversarial attacks on a given sample does provide a piece of information about how far this sample is from the decision boundary. This distance between the decision boundary and the closest sample is indeed relevant in active learning methods (Balcan, Broder, and Zhang 2007), known as margin-based active learning.

In this article, we introduce a new active learning strategy: building on the margin-based learning theory, we leverage adversarial attacks to select unlabeled data to be annotated. Our active strategy is highly innovative as it breaks from the class of uncertainty-based approaches, and is tailored to be most efficient with deep networks. Our contributions are threefold:

• We design a new heuristic for margin-based active learning for deep networks, called DeepFool Active Learning (DFAL). DFAL performs adversarial attacks on the unlabeled samples and selects those which are the closest to their adversarial counterpart. An additional built-in advantage is that the same annotation also labels the adversary, thereby increasing the

training set by two at the cost of one query only.

• Our active learning method leverages the transferability of adversarial examples to transfer the training set from one network to another. Indeed, it has been shown that under some constraints of similarities between the architectures, adversarial examples of a network A are very likely to be adversarial for a network B. This turns to be a significant advantage for our adversarial active learning strategy since the training set built with DFAL for the network A will then be very likely to be a relevant training set for the network B.

• Numerical experiments on a range of real datasets (*MNIST*, *Quick-Draw*, *Shoe-Bag*, *CIFAR10* and *Cats & Dogs*) demonstrate the benefits of our method against the most recent active learning methods for deep networks. For instance, our active learning strategy DFAL queries up to 33% fewer samples than the current state-of-the-art CORE-SET on *MNIST* and *Quick-Draw* for both LeNet5 and VGG8 networks. This gap is even more significant when compared with other techniques. Our method is also faster than CORE-SET with gains of up to 16% in running time.

A review of active learning methods is provided in Sec. II. Our proposed method, DFAL, is described in Sec. III. Experimental results are detailed in Sec. IV. Discussion and Conclusion are provided in Sec. V and VI.

Related Work

For a review of classic active learning methods and their applications, we refer the reader to (Settles 2010). Active learning methods iteratively build the training set: the iterative process alternates between training the classifier on the current labeled training set, and after convergence of the resulting model, query an oracle (usually a human annotator) to label a new set of points. Those new points are meant to yield the maximal accuracy improvement for the given increment in the training set size. They are queried from a pool of unlabeled data given the heuristic in use. Several heuristics coexist as it is impossible to obtain a universal active learning strategy effective for any given task (Dasgupta 2005). Indeed, many existing active learning heuristics have proven to be not effective on CNNs. For example, we empirically noticed in our experiments that uncertainty selection, or uncertainty sampling (Lewis and Gale 1994), may perform worse than passive random selection. Since uncertainty selection consists in querying the annotations for the unlabeled samples which lead to predictions with the lowest confidence, its computational cost is low and its setup simple. It has thus been used for deep networks for various tasks, ranging from sentiment classification to Visual-Question-Answering and Named-Entity-Recognition (Zhou, Chen, and Wang 2010; Lin and Parikh 2017; Yanyao Shen 2018). The CEAL method is an extension of uncertainty selection (Wang et al. 2016): CEAL performs uncertainty selection, but also adds highly confident samples. The labels of these samples are not queried but inferred from the network's predictions (also called pseudo-labeling). CEAL is helpful when the network is generalizing. However, CEAL implies new hyperparameters to threshold the prediction confidence. If such a threshold is poorly tuned, it will corrupt the training set with mistaken labels. Uncertainty selection may also be tailored to network ensemble, either by disagreement over the models (*Query by committee*, (Seung, Opper, and Sompolinsky 1992)) or by sampling through the distribution of the weights (*Bayesian active learning*, (Kapoor et al. 2007)). Recently, (Gal, Islam, and Ghahramani 2016) demonstrated that dropout (and other stochastic regularization schemes) is equivalent to performing inference on the posterior distribution of the weights, enabling to leverage the cost of training and updating multiple models. Thus, dropout allows to sample an ensemble of models at test time: to perform *Dropout Query By Committee* (Ducoffe *et al.* , (Ducoffe and Precioso 2015)) or *Bayesian Active Learning* (Gal *et al.* , (Gal, Islam, and Ghahramani 2016)). Gal *et al.* proceeded with a comparison of several active learning heuristics: among all the strategies, BALD which maximizes the mutual information between the predictions and the model posterior consistently outperforms the other strategies.

(Ozan Sener 2018) define the batch active learning problem as a core-set selection. They minimize the population risk of a model learned on a small labeled subset. To do so, they propose an upper bound with a linear combination of the training error, the generalization error and a third term denoted as the core-set loss. Due to the expressive power of CNNs, the authors argue that the first two terms (training and generalization error) are negligible. Therefore the population risk would mainly be controlled by the core-set loss. The core-set loss consists in the difference between the average empirical loss over the set of points which are already labeled, and the average empirical loss over the entire dataset including unlabeled points. If not considering the labels, the core-set loss is upper bounded with the *covering radius*. Here, we denote by *covering radius*, the maximum distance in the output space between any labeled sample's prediction and any unlabeled sample's prediction. Finally, Sener *et al.* used a mixed integer programming heuristic to minimize at best the *covering radius* of the training set. Thanks to their method, called CORE-SET, they achieve state-of-the-art performance in active learning for image classification. However, the computational cost of their method increases exponentially with the number of training samples.

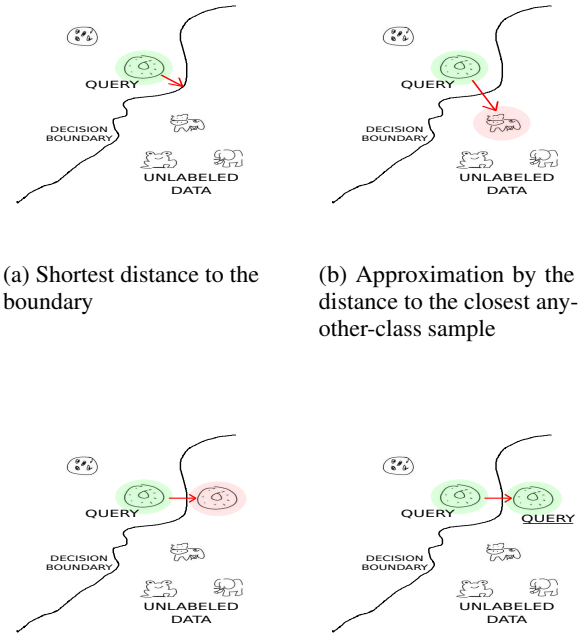
Another direction, rarely explored for deep networks, is to rely on the distance to decision boundaries, namely margin-based active learning. Assuming that the problem is separable with a margin is a reasonable requirement for many popular models: when positive and negative data are separable under SVM, (Tong and Koller 2001) have demonstrated the efficiency of picking the example which is the closest to the decision boundary. Although exploiting the geometric distances has been relevant for active learning on SVM (Tong and Koller 2001; Brinker 2003), it is not straightforward for CNNs since we do not know the geometrical shape of their decision boundaries. One example is the Expected-Gradient-Length strategy from (Zhang, Lease, and Wallace 2017). EGL consists in selecting instances with a high magnitude gradient. Not only such samples will have an impact on the current model parameter estimates, but they will likely modify the shape of the decision boundaries. However, computing the exact gradient for a given sample is intractable without its ground-truth label. In practice, they approximate

the gradient with the expectation over the gradients conditioned on every possible class assignments. In order to reduce the computational cost of EGL approach, the authors consider only the gradients on the embeddings and the output layer.

In this article, we propose a margin-based active strategy where we approximate the margin by the distance between the unlabeled sample the most sensitive to adversarial attacks and its adversarial example.

Adversarial Active Learning

(Balcan, Broder, and Zhang 2007) demonstrated the significant benefit of margin-based approaches in reducing human annotations. We illustrate several margin-based active learning heuristics in figure 1: For each scenario, we query the data highlighted in green. Especially, figure 1(d) describes our contribution. In the original case in figure 1(a), the projection of an unlabeled sample onto the decision boundary determines whether or not it is worth to query its label, depending on the distance between the sample and the boundary. Margin-based strategies are useful, but they require to know how to compute the distance to the decision boundary. When such a distance is intractable, a simple approximation consists in computing the distance between the sample of interest and the closest neighboring sample which has a different prediction.



(a) Shortest distance to the boundary

(b) Approximation by the distance to the closest any-other-class sample

(c) Approximation by the distance to the adversarial example

(d) Our DFAL Strategy

Figure 1: Illustration of different margin-based active learning scenarios in the binary case

Approximating the distance between a sample and the decision boundary, by the distance between this same sample and its closest neighboring sample from a different class, is coarse and computationally expensive.
Instead, we propose DFAL; a DeepFool based Active Learning strategy which selects unlabeled samples with the smallest adversarial perturbation.

Indeed, adversarial attacks were initially designed to approximate the smallest perturbation to cross the decision boundary. Hence, in a binary case, the smallest distance between a sample and its adversarial example better approximates the original distance to the decision boundary than the approximation aforementioned, as illustrated in figure 1(c). Usually, adversarial attacks require the target label. However, in the binary case, the target class of the attack is obvious. In a multi-class setting, things are different: we do not have any prior knowledge on which class the closest adversarial region belongs. Inspired by the strategy done previously in EGL (Zhang, Lease, and Wallace 2017), we could design as many perturbations as the number of classes and keep only the smallest perturbation, but this would be time-consuming. That is why we discard the EGL approach.
We thus have to consider the available techniques of adversarial attacks from the literature (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015; Carlini and Wagner 2016) and look for the most sophisticated and strong attack technique since it will provide more information on the margin. To the best of our knowledge, the method of (Carlini and Wagner 2017) counts among the hardest attacks. However, it also requires to tune several hyperparameters.

We have thus decided to use *DeepFool* algorithm to compute adversarial attacks for DFAL (Moosavi-Dezfooli, Fawzi, and Frossard 2016). Indeed, *DeepFool* is an iterative procedure which alternates between a local linear approximation of the classifier around the source sample and an update of this sample so that it crosses the local linear decision. The algorithm stops when the updated source sample becomes an adversarial sample regarding the initial class of the source sample. When it comes to DFAL, *DeepFool* holds three main advantages: (i) it is hyperparameter free (especially it does not need target labels which makes it more compliant with multi-class contexts); (ii) it runs faster than CORE-SET as we empirically show in table 3; (iii) it is competitive with state-of-the-art adversarial attacks.

To regularize the network and increase its robustness, we add both the least robust unlabeled samples and their adversarial attacks. Thus, it is more likely that the network will regularize on the adversarial examples added to the training set and become less sensitive to small adversarial perturbations. Unlike CEAL, DFAL is hyperparameter-free and cannot corrupt the training set: from the basic definition of adversarial attacks, we know that a sample and its adversarial attack should share the same label.

Finally, DFAL improves the robustness of the network by adding at each iteration unlabeled samples at half the cost of providing their true labels since one label amounts to two samples: the unlabeled sample considered and its adversary. Since the definition of adversarial examples ensures it, we do not consider this trick as pseudo-labeling.

Experiments

Dataset and hyperparameters We evaluate our algorithm for fully supervised image classification on three datasets that have been considered in recent articles on active learning for Deep Learning (Huijser and van Gemert 2017) (table 1): *MNIST*, *Shoe-Bag*, and *Quick-Draw*. For *Quick-Draw*, we downloaded four classes from the Google Doodle dataset: Cat, Face, Angel, and Dolphin.

	img size	**# classes**	**# Training**	**# Test**
MNIST	(28,28,1)	10	60,000	10,000
Shoe-Bag	(64,64,3)	2	184,792	4,000
Quick-Draw	(28,28,1)	4	444,971	111,246
CIFAR10	(64,64,3)	10	???	???
Cats & Dogs	(???)	10	2000	???

Table 1: Summary of the datasets used to evaluate DFAL.

We assess the efficiency of our method on two CNNs: LeNet5 and VGG8 (*Adam, lr=0.001, batch=32*). We use Keras and Tensorflow (Chollet and others 2015; Abadi et al. 2016). Note that DFAL may be used on any architectures impaired by adversarial attacks.

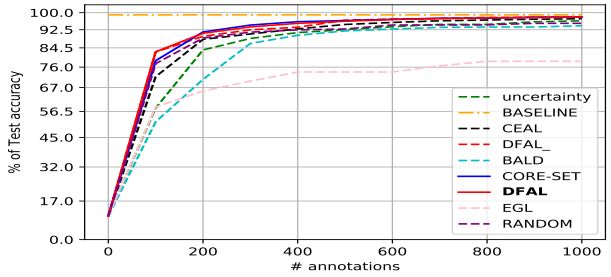
Evaluation We compare the evolution of the test accuracy when using DFAL against the following baselines:

- BALD: we select on a random subset of the unlabeled training set the first n_{query} samples which are expected to maximize the mutual information with the model parameters. In that order, we sample 10 networks from the approximate posterior of the weights by also applying dropout at test time.
- CEAL: we select on the whole unlabeled training set the first n_{query} samples with the highest entropy on their network's prediction. We also label any unlabeled samples whose entropy is lower than a given threshold (which is set according to the authors' guidelines: 0.05 for *MNIST*, 0.19 for *Shoe-Bag* and 0.08 for *Quick-Draw*). Their labels are not queried but estimated from the network's predictions.
- CORE-SET: we select on a random subset of the unlabeled training set the n_{query} samples which cover at best the training set (labeled and unlabeled data) based on the euclidean distance on the output of the last fully connected layer. To approximate the cover set problem, we follow the instructions prescribed in (Ozan Sener 2018): we initialize the selection with the greedy algorithm, and iterate with their mixed integer programming subroutine. We also handle the robustness as prescribed by the authors. We use *or-tools* [1] to reproduce the MIP subroutine.
- EGL: we select from a random subset of the unlabeled training set the first n_{query} samples whose gradients achieves the highest euclidean norm.
- uncertainty: we select from the whole unlabeled training set the first n_{query} samples with the highest entropy on their network's prediction.
- RANDOM: we select randomly from the whole unlabeled training set n_{query} samples.

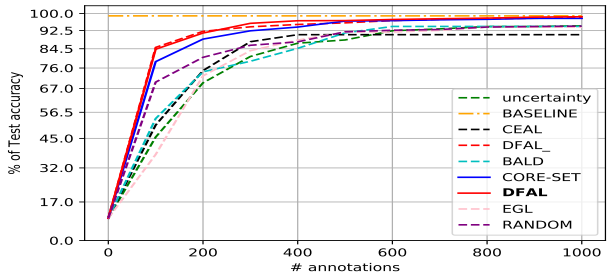
[1] https://developers.google.com/optimization

We average our results over five trials and we plot in figures 2,3,4 the test accuracy achieved by each active learning methods for fixed size training set: with 100, 200, ... to 1000 labeled samples. We denote as *BASELINE*, the test accuracy when training the network on the full labeled training set. First, an interesting observation is that, independently from networks or datasets, active learning methods originally designed for singleton query (BALD, CEAL, EGL, uncertainty) fail to always compete against random selection (fig 4). This may result from the correlations among the queries when using one sample at-a-time. When it comes to our method, DFAL tends to convergence faster than such methods and is always better than random selection, independently from the network or the dataset (fig.2,3,4). Hence our method is more robust to hyperparameter settings than other active learning methods which consider one sample at a time. For various configurations (*Shoe-Bag* with LeNet5 and *Quick-Draw* with VGG8), CEAL is worse than uncertainty selection, hence it selects samples with high entropy but mistaken predictions which add noise into the training set. Unlike CEAL whose probability of acquiring extra samples depends on the efficiency of the network, DFAL holds a constant number of extra queries, depending only on the number of queries. Moreover DFAL creates artificial data which are not part of the pool of data. For example, in tables 2(a) and 2(c), CEAL used more than 20% of the training set of *MNIST* and *Shoe-Bag*, while DFAL only used at most 2%. Thus, DFAL allows more queries, and may also be combined with CEAL. We observe that DFAL always remains in the top three of the best performing active learning methods. We define those methods based on the test error rate when the labeled training set reaches 1000 samples. When DFAL is outperformed, it is only by a really slight percentage of accuracy (at most 0.15%), either by pseudo-labeling method (*which contributes more to the training set*), or by CORE-SET. Since CORE-SET is designed as a batch active learning strategy, it diminishes the correlations among the queries. In order to outperform CORE-SET, DFAL could be extended into a batch setting approach: instead of selecting the top score samples, one could increase the diversity using for example submodular heuristics (Wei, Iyer, and Bilmes 2015). Finally, table 2 compares the effective number of annotations and real number of data required by active learning to reach the same test accuracy than when training on the full labeled training set. We only compare DFAL with the best two active learning methods on 1000 samples. We notice that DFAL always converges with the smallest number of annotations, on *MNIST* and *Quick-Draw*, for both LeNet5 and VGG8 networks: up to 33% less samples than the current state-of-the-art CORE-SET and up to 80% less samples than CEAL. When it comes to *Shoe-Bag*, DFAL remains competitive with CORE-SET and CEAL, overall less than 1% of the training set is needed.

Comparative study between DFAL and CORE-SET In most of our experiments, DFAL is competitive with the current state-of-the-art method, CORE-SET, sometimes outperforming it by a large margin (tab 2(e),2(f)). On the other



(a) LeNet5



(b) VGG8

Figure 2: Evolution of the test accuracy achieved by 7 active learning techniques on *MNIST* given the number of annotations. We denote by DFAL_ our active learning method when not adding the adversarial examples.

	Accuracy $\geq$ 99.04 %	
	# annotations	# labeled data
DFAL	**1210**	2410
CORE-SET	1810	1810
CEAL	$\geq$6000	$\geq$6150

(a) *MNIST*(LeNet5)

	Accuracy $\geq$ 98.98 %	
	# annotations	# labeled data
DFAL	**980**	1950
CORE-SET	1270	1270
uncertainty	2800	2800

(b) *MNIST*(VGG8)

	Accuracy $\geq$ 99.70 %	
	# annotations	# labeled data
DFAL	1070	2130
CORE-SET	**860**	860
CEAL	1130	19157

(c) *Shoe-Bag*(LeNet5)

	Accuracy $\geq$ 99.50 %	
	# annotations	# labeled data
DFAL	530	1050
CORE-SET	**400**	400
CEAL	580	705

(d) *Shoe-Bag*(VGG8)

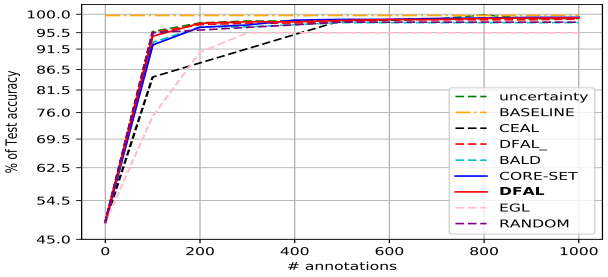
	Accuracy $\geq$ 95.46%	
	# annotations	# labeled data
DFAL	**7470**	14930
CORE-SET	$\geq$8590	$\geq$8590
uncertainty	$\geq$10590	$\geq$10590

(e) *Quick-Draw*(LeNet5)

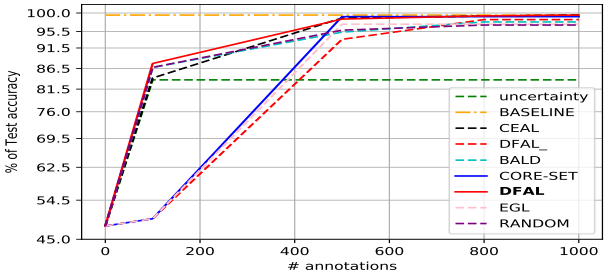
	Accuracy $\geq$ 96.75%	
	# annotations	# labeled data
DFAL	**4810**	9610
CORE-SET	$\geq$6750	$\geq$6750
BALD	5590	5590

(f) *Quick-Draw*(VGG8)

Table 2: Number of annotations to achieve the same test accuracy on LeNet5 and VGG8 as the accuracy obtained on the full training set (BASELINE, $\pm$ 0.5 %).

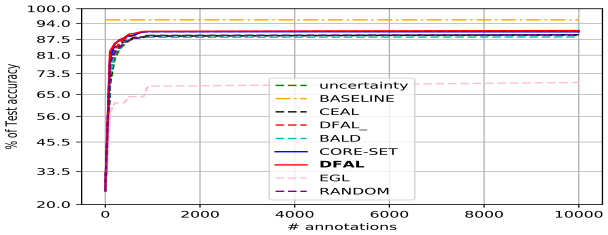


(a) LeNet5

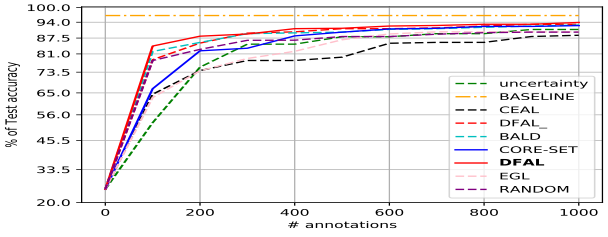


(b) VGG8

Figure 3: Evolution of the test accuracy achieved by 7 active learning techniques on *Shoe-Bag* given the number of annotations. We denote by DFAL_ our active learning method when not adding the adversarial examples.



(a) LeNet5



(b) VGG8

Figure 4: Evolution of the test accuracy achieved by 7 active learning techniques on *Quick-Draw* given the number of annotations. We denote by DFAL_ our active learning method when not adding the adversarial examples.

MNIST	DFAL	CORE-SET (with regularisation)	CORE-SET (no regularisation)
$\|\mathcal{L}\| = 100, \|\mathcal{U}\| = 800$	**126.54**	891.78	784.99
$\|\mathcal{L}\| = 100, \|\mathcal{U}\| = 800$	**126.54**	???	???

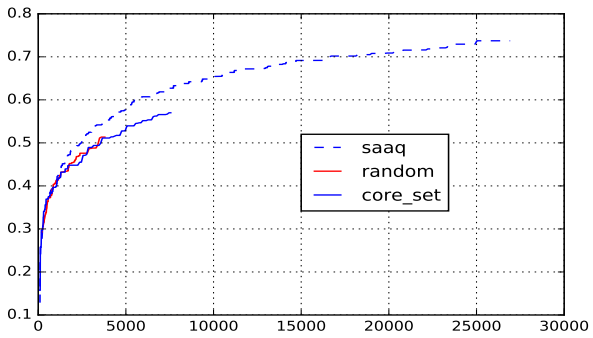
Table 3: Average runtime of DFAL and CORE-SET on *MNIST*. We denote by $\mathcal{L}$ the labeled training set, and $\mathcal{U}$ the unlabeled set of data; $n_{query} = 10$

hand, our method is more interesting than CORE-SET when considering the computational time. DeepFool yields high-performing perturbation vector compared with other state-of-the-art attacks, while being computationally efficient: it converges in a few iterations (less than 3). At each iteration it requires (#classes -1) forward and backward passes. As our DFAL technique uses DeepFool, our active selection criterion is highly efficient compared to the current state-of-the-art CORE-SET. We demonstrate the computational time gap between our method, DFAL, and CORE-SET in table 3: we have recorded the average runtime of selecting 10 queries on *MNIST*. For a sake of fairness, we compare DFAL running time against the CORE-SET approach, with and without robustness [2]. Notice that the runtime performance of DFAL is independent from the size of the labeled training set, while, on the contrary, CORE-SET slows down while we add more and more data to the training set. Eventually, table 3 reports gains of (up to) 16% in running time by our method against CORE-SET. It is worth noting that adversarial attacks are independent, which could easily lead to a parallelized active learning strategy. However, for a fair comparison, with CORE-SET we only consider sequential attack generation.

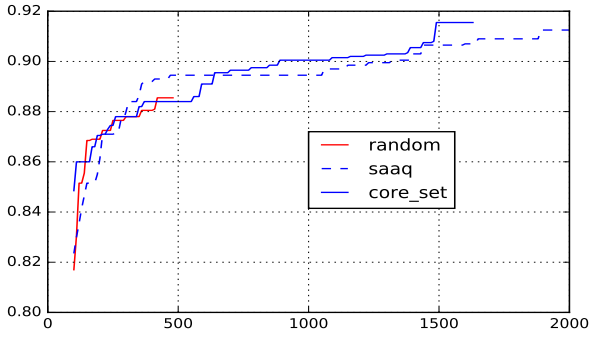
We investigate further the comparison between DFAL and CORE-SET on two experiments: a first experiment studies the behaviour of both active learning methods on a large scale dataset, *CIFAR10* , while a second experiment analyze the combination of DFAL with transfer learning. Firstly, we train a CNN on *CIFAR10* with 5 layers of convolution and 2 fully connected layers with a dropout rate of 0.25. On the full dataset, without artificial augmentation. In figure 5(a), CORE-SET achieves similar accuracy than RANDOM. On the other hand, our method DFAL converges much faster. transfer learning

Transferability When faced with a new classification problem, we know in advance neither the model architecture nor the hyperparameters that are best suited for the problem. One can argue that a network with high capacity is likely to give high accuracy and is sufficient enough when combined with some human expertise on the problem: several architectures have been handcrafted for specific tasks and are available online. Still, their efficiency is known with large datasets. (Yanyao Shen 2018) pointed out an interesting flaw in active learning: they succeed in outperforming classical methods for Named Entity Recognition using only 25% of the training set but by introducing a lightweight architecture. Such an issue is inherent to active learning. Combining model selection

[2]Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz; 64 GB memory and GTX TITAN X



(a) CIFAR



(b) Transfer Learning (Cats & Dogs)

Figure 5: Evolution of the test accuracy achieved by 3 active learning techniques(DFAL, CORE-SET, and RANDOM) given the number of annotations

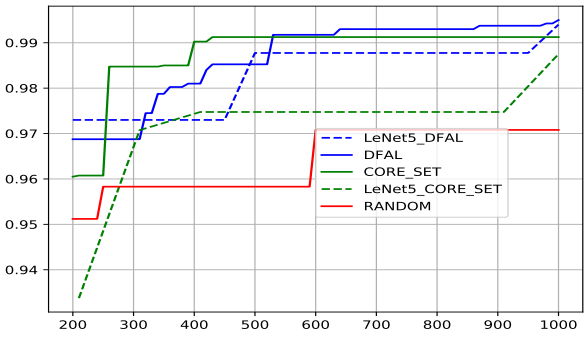


Figure 6: Evolution of the test accuracy for (*Shoe-Bag*, VGG8) trained with different labeled training set: we compare the efficiency of DFAL and CORE-SET built on LeNet5 and transfered on VGG8. The data selected by DFAL for LeNet5 achieve better test accuracy than the data transfered from CORE-SET.

with active learning has been investigated for shallow models. One of the main issue raised is that multiple hypotheses (net-

	DFAL	CORE-SET	RANDOM
LeNet5→ VGG8	**97.80**	96.90	94.46
VGG8→ LeNet5	**97.93**	97.40	95.31

(a) *MNIST*

	DFAL	CORE-SET	RANDOM
LeNet5→ VGG8	**92.87**	91.06	89.94
VGG8→ LeNet5	89.23	89.41	**89.42**

(b) *Quick-Draw*

	DFAL	CORE-SET	RANDOM
LeNet5→ VGG8	**99.40**	99.12	97.08
VGG8→ LeNet5	**98.75**	98.50	98.07

(c) *Shoe-Bag*

Table 4: Comparison of the transferability of DFAL and CORE-SET with 1000 annotations

works) trained in parallel may benefit from labeling different training points. However, deep learning owns a specific property: the transferability of adversarial examples towards a wide range of architectures lead to assuming that the decision borders of neural networks trained on similar tasks overlap.

Based on that argument, we may assume that most of the DFAL queries are useful for a diverse set of architectures, not only the one they have been queried for (considering several architectures but for the same classification problem).

When it comes to the transferability, we empirically demonstrate DFAL potential on a baby task: in figure 6 we recorded *Shoe-Bag*adversarial queries for LeNet5 and use them for training VGG8. While the test accuracy achieved is lower than with the adversarial active queries designed for VGG8, the transfered training set achieves better accuracy than random selection. Also, when reaching 1000 annotated samples, it is better than queries from other active criteria designed for VGG8. We go further and compare the test accuracy of DFAL and CORE-SET transfered dataset on 1000 samples in table 4. Surprisingly the transfered queries from CORE-SET perform better than random. However, in almost every case, the transfered queries from DFAL outperform CORE-SET and RANDOM.

We have therefore shown the relevance of transfering adversarial examples generated within active learning from one architecture to another. This opens up promising perspective for the design of tractable methods to explore network architectures.

Discussion

Theoretical motivations

It is challenging to demonstrate theoretically the gain in annotations of DFAL owing to (i) the high-dimensional space induced by deep networks and (ii) the weak understanding of the phenomenon of adversarial examples. However, we have lately been able to prove the gain of DFAL for linear classifiers. Specifically, we demonstrate the theoretical gain in reducing the labeling effort when data are drawn from the unit ball and consistent with a linear separator with no bias. Our proof is available as a supplementary material.

DFAL does not select random samples

DFAL is very promising empirically. However, for complex network architectures with millions of parameters like VGG8, but trained on a small labeled set, it seems plausible that any example is vulnerable to small adversarial attacks. We clarify this hypothesis and explains why we do not observe such behavior in practice.

Independently of the number of parameters of the network, (Fawzi et al. 2017) have empirically observed that state-of-the-art deep networks learn connected classification regions instead of shattered and disconnected regions. Although such classification regions defined in the input space may suffer from the curse of dimensionality, eventually few directions interfere with the decision boundaries. Considering now the low dimensional space defined by these impacting directions, it becomes likely that the samples do not suffer anymore from the curse of dimensionality and, thus the distance to the decision boundary will differ among the samples. Hence, even in the first iterations of DFAL, we expect the magnitude of the smallest adversarial perturbations to be diverse enough so not to select samples randomly.

Finally, we observe in figure that adversarial perturbations are far from being constant. We believe that the underlying topology of classification regions of deep networks explains the efficiency of our method, even in the first runs.

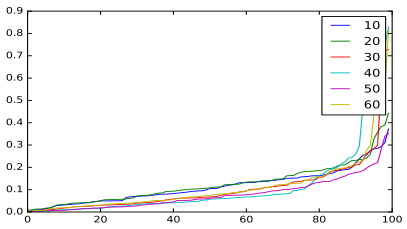


Figure 7: distances between samples and their adversary for VGG8 trained on *MNIST* with 10, 20, 30 to 100 labeled examples. A curve corresponds to the range of adversarial pertubation found on the unlabeled examples, while its color matches the size of the labeled set used to train the network

Conclusion

In this paper, we propose a new heuristic, DFAL, to perform margin-based active learning for CNNs: we approximate the projection of a sample to the decision boundary by its smallest adversarial attack. We demonstrate empirically that our adversarial active learning strategy is highly efficient for CNNs trained on *MNIST*, *Shoe-Bag*, and *Quick-Draw*. Not only we are competitive with the state-of-the-art batch active learning method for CNNs, CORE-SET, but we also outperform CORE-SET for runtime performance. Thanks to the transferability of adversarial attacks, DFAL is a promising approach for combining active learning with model selection for deep networks

References

[Abadi et al. 2016] Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. 2016. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, 265–283.

[Asghar et al. 2017] Asghar, N.; Poupart, P.; Jiang, X.; and Li, H. 2017. Deep active learning for dialogue generation. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, 78–83.

[Balcan, Broder, and Zhang 2007] Balcan, M.-F.; Broder, A.; and Zhang, T. 2007. Margin based active learning. *Learning Theory* 35–50.

[Brinker 2003] Brinker, K. 2003. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, 59–66.

[Carlini and Wagner 2016] Carlini, N., and Wagner, D. 2016. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*.

[Carlini and Wagner 2017] Carlini, N., and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, 39–57. IEEE.

[Chollet and others 2015] Chollet, F., et al. 2015. Keras.

[Dasgupta 2005] Dasgupta, S. 2005. Analysis of a greedy active learning strategy. In *Advances in neural information processing systems*, 337–344.

[Ducoffe and Precioso 2015] Ducoffe, M., and Precioso, F. 2015. Qbdc: Query by dropout committee for training deep supervised architecture. *arXiv preprint arXiv:1511.06412*.

[Fawzi et al. 2017] Fawzi, A.; Moosavi-Dezfooli, S.-M.; Frossard, P.; and Soatto, S. 2017. Classification regions of deep neural networks. *arXiv preprint arXiv:1705.09552*.

[Gal, Islam, and Ghahramani 2016] Gal, Y.; Islam, R.; and Ghahramani, Z. 2016. Deep Bayesian active learning with image data. In *Bayesian Deep Learning workshop, NIPS*.

[Goodfellow, Shlens, and Szegedy 2015] Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. *ICLR 2015*.

[Hoi et al. 2006] Hoi, S. C.; Jin, R.; Zhu, J.; and Lyu, M. R. 2006. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, 417–424. ACM.

[Huijser and van Gemert 2017] Huijser, M. W., and van Gemert, J. C. 2017. Active decision boundary annotation with deep generative models. *arXiv preprint arXiv:1703.06971*.

[Kapoor et al. 2007] Kapoor, A.; Grauman, K.; Urtasun, R.; and Darrell, T. 2007. Active learning with gaussian processes for object categorization. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 1–8. IEEE.

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

[Lin and Parikh 2017] Lin, X., and Parikh, D. 2017. Active learning for visual question answering: An empirical study. *ArXiv e-prints*.

[Moosavi-Dezfooli, Fawzi, and Frossard 2016] Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2574–2582.

[Ozan Sener 2018] Ozan Sener, S. S. 2018. Active learning for convolutional neural networks: A core-set approach. *International Conference on Learning Representations*. accepted as poster.

[Settles 2010] Settles, B. 2010. Active learning literature survey. *University of Wisconsin, Madison* 52(55-66):11.

[Seung, Opper, and Sompolinsky 1992] Seung, H. S.; Opper, M.; and Sompolinsky, H. 1992. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, 287–294. ACM.

[Smith et al. 2018] Smith, J. S.; Nebgen, B.; Lubbers, N.; Isayev, O.; and Roitberg, A. E. 2018. Less is more: sampling chemical space with active learning. *ArXiv e-prints*.

[Szegedy et al. 2014] Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

[Tong and Koller 2001] Tong, S., and Koller, D. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research* 2(Nov):45–66.

[Wang et al. 2016] Wang, K.; Zhang, D.; Li, Y.; Zhang, R.; and Lin, L. 2016. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*.

[Wei, Iyer, and Bilmes 2015] Wei, K.; Iyer, R.; and Bilmes, J. 2015. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, 1954–1963.

[Yanyao Shen 2018] Yanyao Shen, Hyokun Yun, Z. C. L. Y. K. A. A. 2018. Deep active learning for named entity recognition. *International Conference on Learning Representations*. accepted as poster.

[Yu, Bi, and Tresp 2006] Yu, K.; Bi, J.; and Tresp, V. 2006. Active learning via transductive experimental design. In *Proceedings of the 23rd international conference on Machine learning*, 1081–1088. ACM.

[Zhang, Lease, and Wallace 2017] Zhang, Y.; Lease, M.; and Wallace, B. 2017. Active gaminative text representation learning.

[Zhou, Chen, and Wang 2010] Zhou, S.; Chen, Q.; and Wang, X. 2010. Active deep networks for semi-supervised sentiment classification. In *ACL International Conference on Computational Linguistics*, 1515–1523.

[Zhu and Bento 2017] Zhu, J.-J., and Bento, J. 2017. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*.