Adversarial Active Learning for Deep Networks:
a Margin Based Approach

Melanie Ducoffe [1]   Frederic Precioso [1]

Abstract

We propose a new active learning strategy designed for deep neural networks. The goal is to minimize the number of data annotation queried from an oracle during training. Previous active learning strategies scalable for deep networks were mostly based on uncertain sample selection. In this work, we focus on examples lying close to the decision boundary. Based on theoretical works on margin theory for active learning, we know that such examples may help to considerably decrease the number of annotations. While measuring the exact distance to the decision boundaries is intractable, we propose to rely on adversarial examples. We do not consider anymore them as a threat instead we exploit the information they provide on the distribution of the input space in order to approximate the distance to decision boundaries. We demonstrate empirically that adversarial active queries yield faster convergence of CNNs trained on *MNIST*, the *Shoe-Bag* and the *Quick-Draw* datasets.

1. Introduction

The efficiency of deep networks is mainly known under typical training procedures and with large datasets. However, gathering and annotating huge 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 build online an efficient but reduced training set is to rely on active learning. Active learning is a family of methods seeking to optimize automatically the training set for the task at hand in order to limit the need of human annotation. Active learning strategies are not only motivated by theoretical

works demonstrating that one model may perform better using less labeled data if the data are model-crafted (Cohn et al., 1996), but also by its proven efficiency on a wide range of machine learning procedures: from preference rating information for a new user in a movie recommendation system (Sun et al., 2013) to classifying medical data that often requires very high cost labeling (Hoi et al., 2006). It is only recently that active learning has been investigated on deep networks, especially CNNs. The question to scale active learning on deep networks has been raised on a diverse range of topics: from image classification, to sentiment classification, or to VQA and dialogue generation (Gal et al., 2016; Zhou et al., 2010; Lin & Parikh, 2017; Asghar et al., 2017). All those works converge to a common assessment on the efficiency of active learning to reduce the need for a large labeled training set. Yet, transposing directly existing active learning on deep networks is not intuitive. First of all, scaling them to high dimensional parameters networks may turn out to be intractable: some classic active learning methods such as Optimal Experiment Design (Yu et al., 2006) require to inverse the Hessian matrix of the models at each iteration, which would be intractable for current standard CNNs. Secondly, one of the most standard active learning strategy is to rely on uncertainty measure. Uncertainty in deep networks is usually evaluated through the network's output however this is known to be misleading. Indeed, the discovery of adversarial examples has demonstrated that the way we are measuring uncertainty may be overconfident. Adversarial examples are inputs modified with small (sometimes not perceptually distinguishable) but specific perturbations which result in an unexpected misclassification despite a strong confidence of the network in the predicted class(Szegedy et al., 2014). On one hand, the existence of such adversarial examples somehow discards uncertainty-based selection from being an efficient active learning criterion for deep networks. On the other hand, the magnitude of adversarial attacks does provide an information about how far a sample is from the decision boundaries of a deep network. This information is relevant in active learning and known as margin-based active learning. In a generic margin-based active learning, we assume that the decision boundaries evolve towards the optimal solution as the training set increases. Hence samples lying the farthest

*Equal contribution [1]University of Nice Cote dÁzur, - I3S, UMR UNS-CNRS 7271 06900 Sophia Antipolis, France. Correspondence to: Melanie Ducoffe <ducoffe@i3s.unice.fr>.

from the decision boundaries do not need to be labeled by a human expert, as long as the current model is consistent in its predictions with the optimal solution. In order to refine the current model, margin-based active learning queries the unlabeled samples lying close to the decision boundary. Balcan *et al.* , in (Balcan et al., 2007), has demonstrated the significant benefit of margin-based approaches in reducing human annotations: in specific cases, one may obtain an exponential improvement over human labeling. However, it requires computing the distance between a sample and the decision boundaries which is not tractable when considering deep networks. Although we can approximate this distance by considering the minimal distance between two samples from different classification regions (i.e. corresponding to two different classes), such an evaluation is computationally expensive, nor it provides a close upper bound to the real criterion. Eventually, the minimal adversarial perturbation of a sample does provide a better upper bound on how far this sample is from the decision boundaries.

In this article, we do not consider adversarial examples as a threat but rather as a guidance tool to query new data. Our work focuses on a new active selection criterion based on the sensitiveness of unlabeled examples to adversarial attacks. Specifically, our contributions are twofold:

• We present a new heuristic for margin-based active learning for deep networks, called DeepFool Active Learning method (DFAL). It queries the unlabeled samples, which are the closest to their adversarial attacks, labels not only the unlabeled sample but its adversarial counterparts as well, using twice the same label. This pseudo-labeling comes for free without introducing any corrupted labels in the training set.

• We empirically demonstrate that DFAL labeled data may be used on other networks than the one they have been designed for, while achieving higher accuracy than random selection. To the best of our knowledge, this is the first active learning method for deep networks tested for this property.

We describe other active learning methods in the section *Related work*. The following section, *Adversarial Active Learning with Deep-Fool attacks*, describes our method DFAL . Finally, in *Experiments*, we demonstrate empirically the efficiency of our algorithm on three datasets that have been considered in recent methods on active learning for deep networks: *MNIST* , *Quick-Draw* , and *Shoe-Bag* . Not only we achieve state-of-the-art accuracy on those three tasks, but our methods run much faster than the previous state-of-the-art approaches.

2. Related Work

For a review of classic active learning methods and their applications, we refer the reader to Burr Settles (Settles, 2010). The main principle of active learning methods lies in iteratively building the training set: the iterative process alternates between training the classifier on the current labeled training set, and after convergence of the model, asks an oracle (usually a human annotator) to label a new set of points. Those new points 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). When it comes to deep learning, especially CNN, many existing active learning heuristics have proven to be not effective. For example, we empirically noticed in our experiments that uncertainty selection, or uncertainty sampling (Lewis & 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 lowest confidence, its cost is low and its setup simple. It has thus been used on deep networks for various tasks, ranging from sentiment classification to visual question answering and Named Entity Recognition (Zhou et al., 2010; Lin & Parikh, 2017; Yanyao Shen, 2018). Uncertainty selection has been improved in a pseudo-labeling method called CEAL (Wang et al., 2016): CEAL performs uncertainty selection, but also adds highly confident samples into the increased training set. The labels of these samples are not queried but infered from the network's predictions. In the case, one deal with a highly accurate network, CEAL will definitely improve the generalization accuracy. However, CEAL implies new hyperparameters to threshold the prediction's confidence. If such a threshold is badly tuned, it will corrupt the training set with mistaken labels. Uncertainty selection may be also tailored to network ensemble, either by disagreement over the models (*Query by committee*, (Seung et al., 1992)) or by sampling through the distribution of the weights (*Bayesian active learning*, (Kapoor et al., 2007)). Recently, Gal *et al.* , in (Gal et al., 2016), demonstrated that dropout (and other stochastic regularization schemes) is equivalent to perform 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 & Precioso, 2015)) or *Bayesian Active Learning* (Gal *et al.* , (Gal et al., 2016)). Gal *et al.* proceeded with a comparison of several active learning heuristics: among all the metrics, BALD which maximizes the mutual information between predictions and model posterior consistently outperforms other metrics.

In the original problem, active learning only queries one sample at a time. However, such a strategy would not be stable considering deep networks. Since CNNs, and other

deep learning algorithms, are trained with local optimization schemes, we need to add several sample at a time to have a consistent impact on the training. A possible solution is to select the samples with the top scores.

Sener *et al*. (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 equivalent to computing the covering radius over the network prediction. Finally, Sener *et al*. used a mixed integer programming heuristic to minimize at best the covering radius of the data. Thanks to their method, they achieve state-of-the-art performance in active learning for image classification.

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 assumed for many popular models such as SVM, Perceptron or AdaBoost. When positive and negative data are separable under SVM, Tong *et al*. have demonstrated the efficiency of picking the example which is the closest to the decision boundary (Tong & Koller, 2001). If, exploiting the geometric distances has been relevant for active learning on SVM (Tong & Koller, 2001; Brinker, 2003), it is not intuitive for CNNs since we do not know beforehand the geometrical shape of their decision boundaries. A first trial has been proposed in (Zhang et al., 2017). The Expected-Gradient-Length strategy (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 true 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.

3. Adversarial Active Learning with Deep-Fool attacks

In (Balcan et al., 2007), Balcan *et al*. 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, the

data underlined in green will be queried. Especially, figure 1(d) describes our contribution. In the original case in figure 1(a), the projection of an unlabeled sample to 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 effective but they require to know how to compute the distance to the decision boundary. When such a distance is intractable, a naive approximation consists in computing instead the distance between the sample of interest and its closest neighboring sample which has a different predicted class.

(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
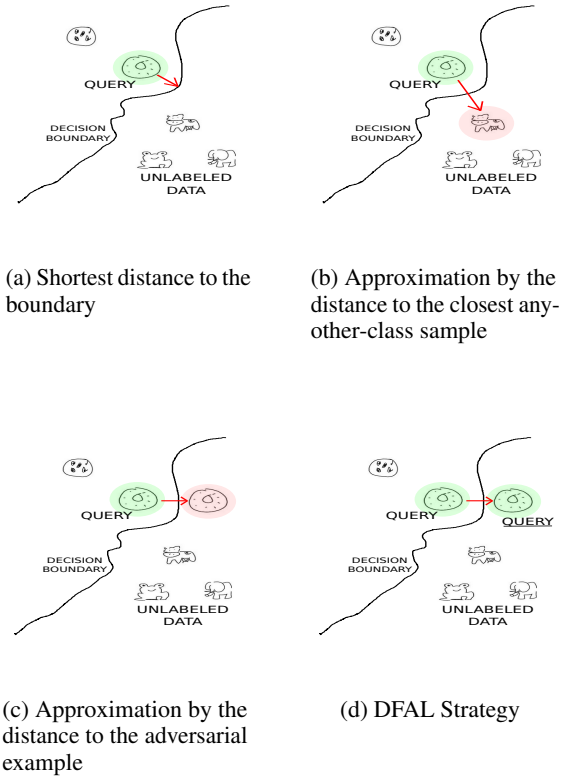
(d) DFAL Strategy

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 Deep-Fool based Active Learning strategy which selects unlabeled samples with the smallest adversarial perturbation.

Indeed, adversarial attacks were originally designed to approximate the smallest perturbation to cross the decision boundary. Hence, in a binary case, the distance between a sample and its smallest adversarial example better approximates the original distance to the decision boundary than the

aforementioned approximation, as illustrated in figure 1(c). In a binary case, the label of the sample added to the training set is then given by the network prediction. Usually, adversarial attacks which would allow us to design a perturbation requires also to know the target label however in a binary case the target class of the attack is obvious.

In a multi-class context everything is different: we do not have any prior knowledge on which class the closest adversarial region belongs to. Inspired from the strategy done previously in EGL (Zhang et al., 2017), we could design as many perturbations as the number of classes and keep only the smallest perturbation, but this would be time consuming. The EGL approach is then discarded.
We thus have to consider the available techniques of adversarial attacks from the literature (Szegedy et al., 2014; Goodfellow et al., 2015; Carlini & Wagner, 2016) and look for the hardest technique to counter since it will provide more information on the margin in more cases and in more difficult cases. To the best of our knowledge, Carlini *et al.* (Carlini & Wagner, 2017b; He et al., 2017; Carlini & Wagner, 2017a) methods are among the hardest attacks to counter. However, it also requires to tune several hyperparameters.

We have thus decided to use *Deep-Fool* algorithm to compute adversarial attacks for DFAL (Moosavi-Dezfooli et al., 2016). Indeed, *Deep-Fool* 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 effectively an adversarial sample regarding the initial class of the source sample. When it comes to DFAL , *Deep-Fool* 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 fast as we empirically noticed in table 3; (iii) it is competitive with state-of-the-art adversarial attacks.

Moreover, DFAL is theoretically motivated by the robustness of neural networks: in (Tom Zahavy, 2018), Xu *et al.* used robustness to explain the generalization abilities of stochastic algorithms. They can generalize well as long as their sensitiveness to adversarial examples is bounded in average. Xu *et al.* explain that since deep learning methods, in the majority of cases, are involving stochastic optimization mechanisms due to the common schemes used in their training phase such as SGD or dropout, they can be considered as stochastic algorithms. Therefore, by adding samples sensitive to small perturbations, DFAL enforces the network to increase its ensemble robustness and generalization abilities.

In order to regularize the network and increase its robustness in DFAL , we add both the less 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 reading their true labels (one label amounts to two samples) as described in Algorithm 1.

Algorithm 1 *DFAL*: DeepFool Active Learning

Require: $\mathcal{L}$ set of initial labeled training examples
Require: $\mathcal{U}$ set of initial unlabeled training examples
Require: $\mathcal{H}$ set of hyper-parameters to train the network
Require: K the number of candidates
Require: n_{query} the number of data to query
Require: p: the L_p norm used ($p = 2$)
Require: N: the number of data to label
  # init the training set
  $k = 0$
  $\mathcal{L}_0 = \mathcal{L}$
  $\mathcal{U}_0 = \mathcal{U}$
  while k<N **do**
    # Train the network $\mathcal{A}_k$ given the current labeled training set
    $\mathcal{A}_k$=training($\mathcal{H}, \mathcal{L}_k$)
    # Select randomly a pool of data $\mathcal{S}_k$ of size K
    $\mathcal{S}_k \subseteq \mathcal{U}_k; \ |\mathcal{S}_k| = K$
    for $x_i \in \mathcal{S}_k$ **do**
      #compute adversarial attacks with L_p norms
      $r_i \leftarrow DeepFool(x_i, \mathcal{A}_k; p)$
    end for
    # query the labels of the n_{query}-th samples $\mathcal{Q}_k$ owing the smallest L_p norm perturbation
    $index_k \leftarrow argsort(<r_i, r_i>_p| \ i = 1..K)$
    $\mathcal{Q}_k \leftarrow \{x_j \mid j \in index_k[0 : n_{query}]\} \cup \{x_j + r_j \mid j \in index_k[0 : n_{query}]\}$
    $\mathcal{L}_{k+1} \leftarrow \mathcal{L}_k \cup \mathcal{Q}_k$
    $\mathcal{U}_{k+1} \leftarrow \mathcal{U}_k \setminus \mathcal{Q}_k$
  end while

---

4. Experiments

4.1. Dataset and CNN

We tested our algorithms for fully supervised image classification on three datasets that have been considered in recent articles on active learning for Deep Learning (Huijser & van Gemert, 2017): *MNIST* , *Shoe-Bag* , and *Quick-Draw* :
• *MNIST* : 28x28 grayscale images from 10 digits classes. The training and test set contains respectively 60,000 and 10,000 samples.

- *Shoe-Bag* : This dataset has been created in (Huijser & van Gemert, 2017) from the Handbags and the Shoes datasets. It contains RGB images of size 64x64: 184,792 for training along with 4,000 images for testing.
- *Quick-Draw* : 28x28 grayscale images from the Google Doodle dataset. We downloaded four classes: Cat, Face, Angel, and Dolphin. This lead us to a training set of 444,971 samples and a test set of size 111,246 samples.

We assess the efficiency of our method on two CNNs: LeNet5 and VGG8 (*Adam, lr=0.001, batch=32*). We have used Keras and Theano (Chollet et al., 2015; Al-Rfou et al., 2016). Although we have only tested our methods for CNNs trained with cross-entropy, DFAL may be used on any architectures impaired by adversarial attacks.

4.2. Evaluation

We compare the evolution of the test accuracy when querying data with DFAL against the following baselines:

1. 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 as test time.

2. 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.

3. 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 have used *or-tools* [1] to reproduce the MIP subroutine.

4. EGL : we select from a random subset of the unlabeled training set, the first n_{query} samples whose gradients

---

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

achieves the highest euclidean norm.

5. uncertainty : we select from the whole unlabeled training set, the first n_{query} samples with the highest entropy on their network's prediction.

6. RANDOM : we select randomly from the whole unlabeled training set n_{query} samples.

We average our results over five trials and plot the accuracy on the test set in figure 2. Also, we index in table 1 the test accuracy achieved by each active learning methods for fixed size training set: with 100, 500, 800, and 1000 labeled samples.

First of all, 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 2). This may result from the correlations among the queries when using top score selection. 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 (table 1). Hence our method is more robust to the hyperparameters settings than other active learning methods, when considering top score selection.

On diverse 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 adds 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 1(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, 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

	Accuracy (%)				
# annotations	100	500	800	1000	All
DFAL	**82.77**	96.23	**97.71**	98.02	–
BALD	51.88	91.96	93.69	94.24	–
CEAL	71.81	94.81	96.77	97.33	–
CORE-SET	78.86	**96.52**	97.53	**98.03**	–
EGL	58.44	73.86	78.57	78.57	–
uncertainty	57.96	92.52	94.84	96.41	–
RANDOM	77.56	92.83	94.63	95.31	**99.04**

(a) *MNIST* (LeNet5)

	Accuracy (%)				
# annotations	100	500	800	1000	All
DFAL	**84.28**	**96.90**	**97.98**	**98.59**	–
BALD	53.73	91.47	94.32	94.32	–
CEAL	50.87	90.69	90.69	90.69	–
CORE-SET	78.80	96.68	97.46	97.88	–
EGL	37.92	91.84	93.99	93.99	–
uncertainty	45.57	88.36	94.27	94.60	–
RANDOM	69.79	91.96	94.05	94.46	**98.98**

(b) *MNIST* (VGG8)

	Accuracy (%)				
# annotations	100	500	800	1000	All
DFAL	94.62	98.50	98.98	99.10	–
BALD	93.10	97.95	97.95	97.95	–
CEAL	84.65	98.50	99.00	99.12	–
CORE-SET	92.50	**98.75**	**99.07**	**99.25**	–
EGL	75.07	95.47	95.47	95.47	–
uncertainty	**95.78**	98.35	98.85	98.98	–
RANDOM	95.50	98.07	98.07	98.07	**99.70**

(c) *Shoe-Bag* (LeNet5)

	Accuracy (%)				
# annotations	100	500	800	1000	All
DFAL	**87.73**	98.53	**99.30**	99.50	–
BALD	86.78	95.35	97.83	97.83	–
CEAL	84.20	98.78	99.25	**99.52**	–
CORE-SET	0.50	**99.12**	99.12	99.12	–
EGL	0.50	97.28	97.28	97.28	–
uncertainty	83.75	83.75	83.75	83.75	–
RANDOM	86.78	95.83	97.08	97.08	**99.50**

(d) *Shoe-Bag* (VGG8)

	Accuracy (%)				
# annotations	100	500	800	1000	All
DFAL	**82.56**	**89.63**	**90.72**	**91.09**	–
BALD	72.65	87.18	88.34	88.45	–
CEAL	70.46	87.04	88.31	89.39	–
CORE-SET	79.58	88.93	90.54	90.53	–
EGL	57.48	64.05	64.05	69.85	–
uncertainty	69.24	86.89	88.54	89.09	–
RANDOM	78.09	87.03	88.98	89.42	**95.46**

(e) *Quick-Draw* (LeNet5)

	Accuracy (%)				
# annotations	100	500	800	1000	All
DFAL	**84.23**	**91.52**	**93.16**	**93.91**	–
BALD	82.00	89.94	91.92	92.87	–
CEAL	64.45	79.66	85.73	88.65	–
CORE-SET	66.71	89.93	92.28	92.62	–
EGL	63.12	86.80	90.06	90.06	–
uncertainty	52.77	88.05	89.31	91.03	–
RANDOM	78.28	88.13	89.71	89.94	**96.75**

(f) *Quick-Draw* (VGG8)

Table 1. Test accuracy achieved by 7 active learning techniques for different number of annotations on LeNet5 and VGG8 .
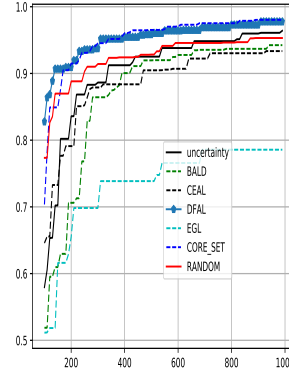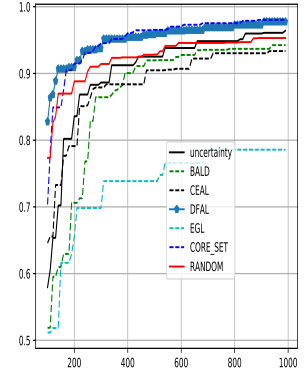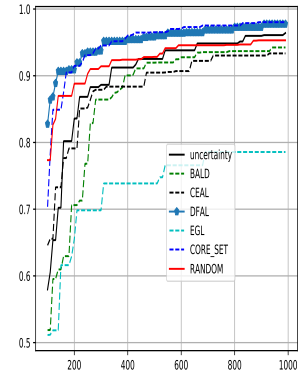

(a) *MNIST* (LeNet5)   (b) *MNIST* (VGG8)
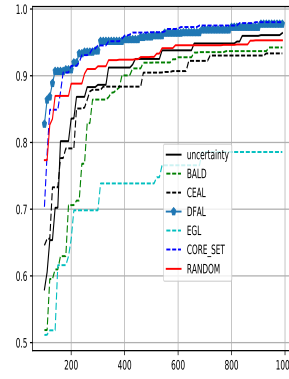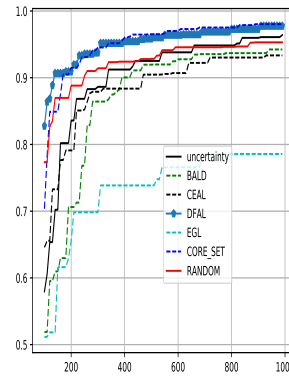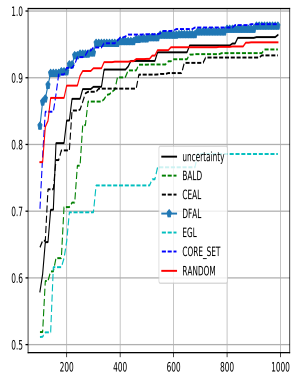
(c) *Shoe-Bag* (LeNet5)   (d) *Shoe-Bag* (VGG8)

(e) *Quick-Draw* (LeNet5)   (f) *Quick-Draw* (VGG8)

Figure 2. Evolution of the test accuracy achieved by 7 active learning techniques(DFAL , BALD , CEAL , EGL , uncertainty and RANDOM) given the number of annotations

	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. Comparison of the number of annotations and effective data required to achieve the same test accuracy on LeNet5 and VGG8 as the accuracy obtained on the full training set ($\pm$ 0.5). We considered DFAL against the active methods achieving best accuracy on 1000 samples.

et al., 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. Regarding top score approaches, we notice that DFAL always converges with the smallest number of annotations, on *MNIST* and *Quick-Draw* . When it comes to *Shoe-Bag* , DFAL remains competitive with the core-set approach and CEAL , overall less than 1% of the training set is needed.

4.3. Comparative study between DFAL and the CORE-SET approach

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(a),2(b)). On the other hand, our method is more interesting than CORE-SET when considering the computational time. Indeed one of the main cons raised against CORE-SET is that the optimal solution is a NP-Hard problem. To overcome this issue, the authors used a greedy solution, which is known to hold a 2-OPT bound. Then, they optimize this solution, using a Mixed Integer Programming subroutine on which they iterate to improve the coverage. While constructing this MIP, they also handle the weakness of k-center, namely robustness: they assume an upper limit on the number of outliers. However, using robustness, as prescribed in the original paper, slows down the active selection. Their solution selects a batch of data at each time, while our method

	DFAL	CORE-SET (with regularisation)	CORE-SET (no regularisation)
second	**126.54**	891.78	784.99

Table 3. Average runtime of DFAL and CORE-SET on *MNIST* : $\mid \mathcal{L} \mid = 100$, $\mid \mathcal{U} \mid = 800$, $n_{query} = 10$

attributes scores to each unlabeled sample independently one from another. Hence DFAL can be easily parallelized to compute adversarial attacks for a large pool of unlabeled samples.

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* with a training set of 100 samples and an unlabeled pool of size 800. 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 CORE-SET slows down while we add more and more data to the training set.
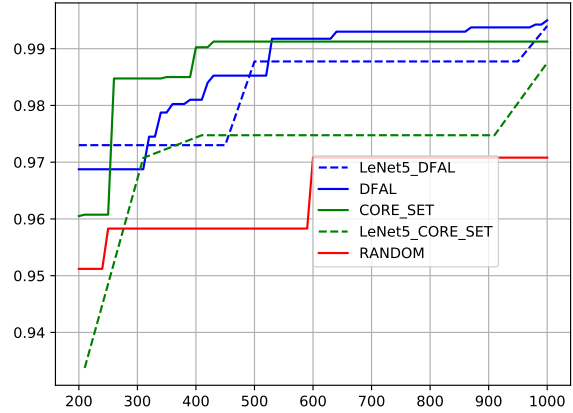
4.4. Transferability

Figure 3. 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 . At 1000 samples, they converge to similar test accuracy than the data specifically designed for VGG8 .

In preliminary experiments to a new 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

---

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

	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

expertise on the problem: several architectures have been handcrafted for specific tasks and are available online. Still, their efficiency is known under typical training procedures and with large datasets. In (Yanyao Shen, 2018), Yanyao Shen *et al*. 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. Hence, when using a single predefined model, active learning may optimize the training set to a model not well optimized for the task at hand. Such an issue is inherent to active learning. Combining model selection with active learning has been investigated for shallow models. One of the main issue raised is that multiple hypotheses trained in parallel may benefit from labeling different training points. Hence an active learning strategy effective on any fixed model may be less efficient than random sampling when considering it with model selection. Although combining model selection and active learning for any type of model is non-trivial, deep learning owns a specific property: the transferability of adversarial examples towards a wide range of architectures lead to assume that the decision borders of neural networks trained on similar tasks overlap.

DFAL overcomes this limitation. Indeed it is well known that adversarial attacks handcrafted for a specific network may be used with success on other networks, especially when considering CNNs. The reason raised is that the distance between network's decision borders is smaller than most adversarial perturbations. 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.

When it comes to the transferability, we empirically demon-

strate DFAL 's potential on a baby task: in figure 3 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, but also, when reaching 1000 annotated samples, it is also 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 . The only exception concerns the transfered queries from VGG8 to LeNet5 : neither DFAL nor CORE-SET succeed in outperforming RANDOM . We believe that LeNet5 trained on *Quick-Draw* have a smoother decision boundary than VGG8 in our hyperparameter setting. Thus, it would result in VGG8 queries being useful for training LeNet5 , while the opposite would not be true.

5. 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

Al-Rfou, Rami, Alain, Guillaume, Almahairi, Amjad, Angermueller, Christof, Bahdanau, Dzmitry, Ballas, Nicolas, Bastien, Frédéric, Bayer, Justin, Belikov, Anatoly, Belopolsky, Alexander, et al. Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 472:473, 2016.

Asghar, Nabiha, Poupart, Pascal, Jiang, Xin, and Li, Hang. Deep active learning for dialogue generation. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pp. 78–83, 2017.

Balcan, Maria-Florina, Broder, Andrei, and Zhang, Tong. Margin based active learning. *Learning Theory*, pp. 35–50, 2007.

Brinker, Klaus. Incorporating diversity in active learning

with support vector machines. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 59–66, 2003.

Carlini, Nicholas and Wagner, David. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

Carlini, Nicholas and Wagner, David. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pp. 3–14, New York, NY, USA, 2017a. ACM. ISBN 978-1-4503-5202-4. doi: 10.1145/3128572.3140444. URL http://doi.acm.org/10.1145/3128572.3140444.

Carlini, Nicholas and Wagner, David. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 39–57. IEEE, 2017b.

Chollet, François et al. Keras, 2015.

Cohn, David A, Ghahramani, Zoubin, and Jordan, Michael I. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.

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

Ducoffe, Melanie and Precioso, Frederic. Qbdc: Query by dropout committee for training deep supervised architecture. *arXiv preprint arXiv:1511.06412*, 2015.

Gal, Yarin, Islam, Riashat, and Ghahramani, Zoubin. Deep Bayesian active learning with image data. In *Bayesian Deep Learning workshop, NIPS*, 2016.

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

He, Warren, Wei, James, Chen, Xinyun, Carlini, Nicholas, and Song, Dawn. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, Vancouver, BC, 2017. USENIX Association. URL https://www.usenix.org/conference/woot17/workshop-program/presentation/he.

Hoi, Steven CH, Jin, Rong, Zhu, Jianke, and Lyu, Michael R. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, pp. 417–424. ACM, 2006.

Huijser, Miriam W and van Gemert, Jan C. Active decision boundary annotation with deep generative models. *arXiv preprint arXiv:1703.06971*, 2017.

Kapoor, Ashish, Grauman, Kristen, Urtasun, Raquel, and Darrell, Trevor. Active learning with gaussian processes for object categorization. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8. IEEE, 2007.

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

Lin, X. and Parikh, D. Active Learning for Visual Question Answering: An Empirical Study. *ArXiv e-prints*, November 2017.

Moosavi-Dezfooli, Seyed-Mohsen, Fawzi, Alhussein, and Frossard, Pascal. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582, 2016.

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

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

Seung, H Sebastian, Opper, Manfred, and Sompolinsky, Haim. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294. ACM, 1992.

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

Sun, Mingxuan, Li, Fuxin, Lee, Joonseok, Zhou, Ke, Lebanon, Guy, and Zha, Hongyuan. Learning multiple-question decision trees for cold-start recommendation. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 445–454. ACM, 2013.

Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, and Fergus, Rob. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

Tom Zahavy, Bingyi Kang, Alex Sivak Jiashi Feng Huan Xu Shie Mannor. Ensemble robustness and generalization of stochastic deep learning algorithms, 2018. URL https://openreview.net/forum?id=r1YUtYx0-.

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

Wang, Keze, Zhang, Dongyu, Li, Ya, Zhang, Ruimao, and Lin, Liang. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

Wei, Kai, Iyer, Rishabh, and Bilmes, Jeff. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pp. 1954–1963, 2015.

Yanyao Shen, Hyokun Yun, Zachary C. Lipton Yakov Kronrod Animashree Anandkumar. Deep active learning for named entity recognition. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ry018WZAZ. accepted as poster.

Yu, Kai, Bi, Jinbo, and Tresp, Volker. Active learning via transductive experimental design. In *Proceedings of the 23rd international conference on Machine learning*, pp. 1081–1088. ACM, 2006.

Zhang, Ye, Lease, Matthew, and Wallace, Byron. Active gaminative text representation learning. 2017.

Zhou, Shusen, Chen, Qingcai, and Wang, Xiaolong. Active deep networks for semi-supervised sentiment classification. In *ACL International Conference on Computational Linguistics*, pp. 1515–1523, 2010.