Bayesian Generative Active Deep Learning

Toan Tran [1]   Thanh-Toan Do [2]   Ian Reid [1]   Gustavo Carneiro [1]

Abstract

Deep learning models have demonstrated outstanding performance in several problems, but their training process tends to require immense amounts of computational and human resources for training and labeling, constraining the types of problems that can be tackled. Therefore, the design of effective training methods that require small labeled training sets is an important research direction that will allow a more effective use of resources. Among current approaches designed to address this issue, two are particularly interesting: data augmentation and active learning. Data augmentation achieves this goal by artificially generating new training points, while active learning relies on the selection of the "most informative" subset of unlabeled training samples to be labelled by an oracle. Although successful in practice, data augmentation can waste computational resources because it indiscriminately generates samples that are not guaranteed to be informative, and active learning selects a small subset of informative samples (from a large un-annotated set) that may be insufficient for the training process. In this paper, we propose a Bayesian generative active deep learning approach that combines active learning with data augmentation – we provide theoretical and empirical evidence (MNIST, CIFAR-$\{10, 100\}$, and SVHN) that our approach has more efficient training and better classification results than data augmentation and active learning.

1. Introduction

Deep learning is undoubtedly the dominant machine learning methodology (Esteva et al., 2017; Huang et al., 2017; Kumar et al., 2016; Rajkomar et al., 2018). Part of the reason behind this success lies in its training process that can be performed with immense and carefully labeled data sets, where the larger the data set, the more effective the training process (Sun et al., 2017). However, the labeling of such large data sets demands significant human effort, and the large-scale training process requires considerable computational resources (Sun et al., 2017). These training issues have prevented researchers and practitioners from solving a wider range of classification problems, where large labeled data sets are hard to acquire or vast computational resources are unavailable (Litjens et al., 2017). Addressing these issues is one of the most important problems to be solved in machine learning (Bengio, 2012; Gal et al., 2017; Kingma et al., 2014; Settles, 2012; Krizhevsky et al., 2012; Tran et al., 2017; Zhu & Bento, 2017).

One of the most successful approaches to mitigate the issue described above relies on the use of a small labeled data set and a large unlabeled data set, where small subsets from the unlabeled set are automatically selected using an acquisition function that assesses how informative those subsets are for the training process. These selected unlabeled subsets are then labeled by an oracle (i.e., a human annotator), integrated into the labeled data set, which is then used to re-train the model in an iterative training process. This algorithm is known as (pool-based) active learning (Settles, 2012), and its aim is to reduce the need for large labeled data sets and the computational requirements for training models because it tends to rely on smaller training sets. Although effective in general, active learning may overfit the informative training sets due to their small sizes.

Alternatively, if the unlabeled data set does not exist, then a possible idea is to use the samples from the labeled set to guide the generation of new artificial training points by sampling from a generative distribution that is assumed to have a particular shape (e.g., Gaussian noise around rigid deformation parameters from the labels) (Krizhevsky et al., 2012) or that have been estimated from a generative adversarial training (Tran et al., 2017). This training process is known as data augmentation, which targets the reduction of the need for large labeled training sets. However, given that the generation of new samples is done without regarding how informative the new sample is for the training process, it is likely that a large proportion of the generated samples will not be important for the training process. Consequently, data augmentation wastes computational resources, forcing

[1]University of Adelaide, Australia [2]University of Liverpool. Correspondence to: Toan Tran <toan.m.tran@adelaide.edu.au>.

the training process not only to take longer than necessary, but also to be relatively ineffective, particularly at the later stages of the training process, when most of the generated points are likely to be uninformative.

In this paper, we propose a new Bayesian generative active deep learning method that targets the augmentation of the labeled data set with generated samples that are informative for the training process – see Fig. 1. Our paper is motivated by the following works: query by synthesis active learning (Zhu & Bento, 2017), Bayesian data augmentation (Tran et al., 2017), auxiliary-classifier generative adversarial networks (ACGAN) (Odena et al., 2017) and variational autoencoder (VAE) (Kingma & Welling, 2013). We assume the existence of a small labeled and a large unlabeled data set, where we use the concept of Bayesian active learning by disagreement (BALD) (Gal et al., 2017; Houlsby et al., 2011) to select informative samples from the unlabeled data set. These samples are then labeled by an oracle and processed by the VAE-ACGAN to produce new artificial samples that are as informative as the selected samples. This set of new samples are then incorporated into the labeled data set to be used in the next training iteration.

Compared to a recently proposed generative adversarial active learning (Zhu & Bento, 2017), which relies on an optimization problem to generate new samples (this optimization balances sample informativeness with image generation quality), our approach has the advantage of using acquisition functions that have proved to be more effective (Gal et al., 2017) than the simple information loss in (Zhu & Bento, 2017). Different from our approach that trains the generative and classification models jointly, the approach in (Zhu & Bento, 2017) relies on a 2-stage training, where the generator training is independent of the classifier training. A potential disadvantage of our method is the fact that the whole unlabeled data set needs to be processed by the acquisition function at each iteration, but that is mitigated by the fact that we can sample a much smaller (fixed-size) subset of the unlabeled data set to guarantee the informativeness of the selected samples (He & Ma, 2013). An important question about the VAE-ACGAN generation process is how informative the generated artificial sample is, when compared with the active learning selected sample from the unlabeled training set. We show that this generated sample is theoretically guaranteed to be informative, given a couple of assumptions that are empirically verified. We run experiments which show that our proposed Bayesian generative active deep learning is advantageous in terms of training efficiency and classification performance, compared with data augmentation and active learning on MNIST, CIFAR-$\{10, 100\}$ and SVHN.

2. Related Work

2.1. Bayesian Active Learning

In a (pool-based) active learning framework, the learner is initially modeled with a small labeled training set, and it will iteratively search for the "most informative" samples from a large unlabeled data set to be labeled by an oracle – these newly labeled samples are then used to re-model the learner. The information value of an unlabeled sample is estimated by an acquisition function, which is maximized in order to select the most informative samples. For example, the most informative samples can be selected by maximizing the "expected informativeness" (MacKay, 1992), or minimizing the "expected error" of the learner (Cohn et al., 1996) – such acquisition functions are hard to optimize in deep learning because they require the estimation of the inverse of the Hessian computed from the expected error with respect to high-dimensional parameter vectors.

Recently, Houlsby et al. (2011) proposed the Bayesian active learning by disagreement (BALD) scheme in which the acquisition function is measured by the "mutual information" of the training sample with respect to the model parameters. Gal et al. (2017) pointed out that, in deep active learning, the evaluation of this function is based on model uncertainty, which in turn requires the approximation of the posterior distribution of the model parameters. These authors also introduced the use of Monte Carlo (MC) dropout method (Gal & Ghahramani, 2016) to approximate this and other commonly used acquisition functions. This approach (Gal et al., 2017) is shown to work well in practice despite the poor convergence of the MC approximation. In our proposed approach, we also use this method to approximate the BALD acquisition function in the active selection process.

2.2. Data Augmentation

In active learning, it is assumed that a model can be trained with a small data set. That assumption is challenging in the estimation of a deep learning model since it often requires large labeled data sets to avoid over-fitting. One reasonable way to increase the labeled training set is with data augmentation that artificially generates new synthetic training samples (Krizhevsky et al., 2012). Gal et al. (2017) also emphasized the importance of data augmentation for the development of deep active learning. Data augmentation can be performed with "label-preserving" transformations (Krizhevsky et al., 2012; Simard et al., 2003; Yaeger et al., 1996) – this is known as "poor's man" data augmentation (PMDA) (Tanner, 1991; Tran et al., 2017). Alternatively, Bayesian data augmentation (BDA) trains a deep generative model (using the training set), which is then used to produce new artificial training samples (Tran et al., 2017). Compared to PMDA, BDA has been shown to have a better theoretical foundation and to be more beneficial in
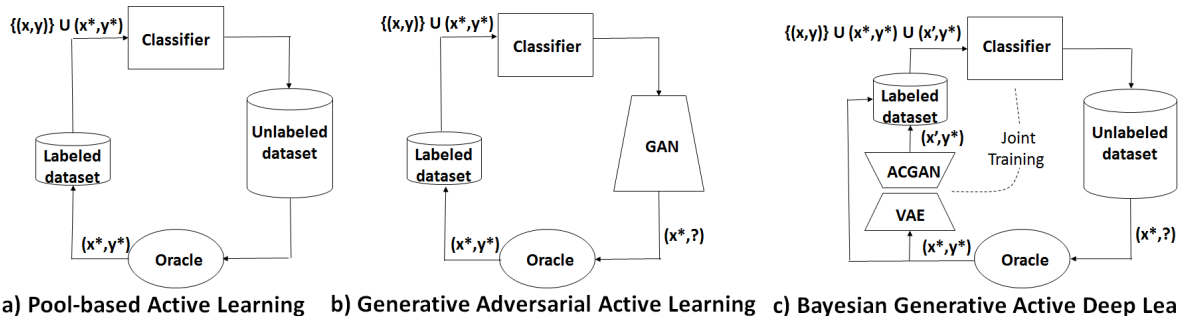
Figure 1. Comparison between (pool-based) active learning (Settles, 2012) (a), generative adversarial active learning (Zhu & Bento, 2017) (b), and our proposed Bayesian generative active deep learning (c). The labeled data set is represented by $\{(\mathbf{x}, \mathbf{y})\}$, the unlabeled point to be labeled by the oracle is denoted by $\mathbf{x}^*$ (oracle's label is $\mathbf{y}^*$), and the point generated by the VAE-ACGAN model is denoted by $\mathbf{x}'$.

practice (Tran et al., 2017). One of the drawbacks of data augmentation is that the generation of new training points is driven by the likelihood that the generated samples belong to the training set – this implies that the model produces samples that are likely to be close to the generative distribution mode. Unfortunately, as the training process progresses, these points are the ones more likely to be correctly classified by classifier, and as a result they are not informative. The combination of active learning and data augmentation proposed in this paper addresses the issue above, where the goal is to continuously generate informative training samples that not only are likely to belong to the learned generative distribution, but are also informative for the training process – see Fig. 2.

2.3. Generative Active Learning

The training process in active learning can be significantly accelerated by actively generating informative samples. Instead of querying most informative instances from an unlabeled pool, Zhu & Bento (2017) introduced a generative adversarial active learning (GAAL) model to produce new synthetic samples that are informative for the current model. The major advantage of their algorithm is that it can generate rich representative training data with the assumptions that the GAN model has been pre-trained and the optimization during generation is solved efficiently. Nevertheless, this approach has a couple of limitations that make it challenging to be applied in deep learning. First, the acquisition function must be simple to compute and optimize (e.g., distance to classification hyper-plane) because it will be used by the generative model during the sample generation process – such simple acquisition functions have been shown to be not quite useful in active learning (Gal et al., 2017). Also, the GAN model in (Zhu & Bento, 2017) is not fine-tuned as training progresses since it is pre-trained only once before generating new instances – as a result, the generative and discriminative models do not "co-evolve".

In contrast, following the standard active learning, our Bayesian Generative Active Deep Learning first queries

the unlabeled data set samples based on their "information content", and conditions the generation of a new synthetic sample on this selected sample. Moreover, the learner and the generator are jointly trained in our approach, allowing them to "co-evolve" during training. We show empirically that, in our proposed approach, a synthetic sample generated from the most informative sample belongs to its sufficiently small neighborhood. More importantly, the value of the acquisition function at the generated sample is mathematically shown to be closed to its optimal value (at the most informative sample), and the synthetic instance, therefore, can also be considered to be informative.

2.4. Variational Autoencoder Generative Adversarial Networks

Generative Adversarial Network (GAN) (Goodfellow et al., 2014) is one of the most studied deep learning models. GANs typically contain two components: a generator that learns to map a latent variable to a sample data, and a discriminator that aims to guide the generator to produce realistically looking samples. The generative performance of GAN is often evaluated by both the quality and the diversity of the synthetic instances. There have been several extensions proposed to improve the quality of the GAN generated images, such as CGAN (Mirza & Osindero, 2014) and ACGAN (Odena et al., 2017). In order to tackle the low diversity problem (known as "mode collapse"), Larsen et al. (2016) introduced a variational autoencoder generative adversarial network (VAE-GAN) that combines a variational autoencoder (VAE) (Kingma & Welling, 2013) and a GAN in which these networks are connected by a generator/decoder (Zhang et al., 2017). We utilize both ACGAN and VAE-GAN in our proposed Bayesian Generative Active Deep Learning framework, but with the aim of improving the classification performance.
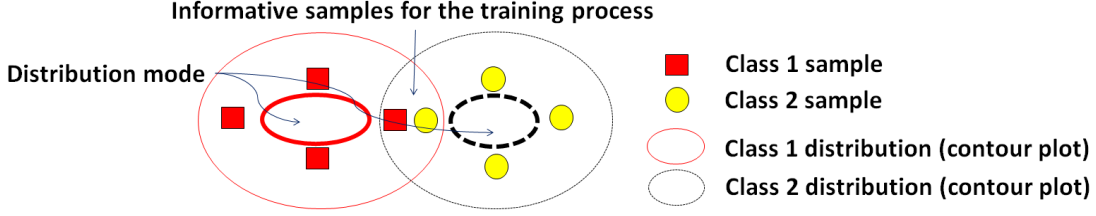
Figure 2. We target the generation of samples that belong to the generative distribution learned from the training set, and that are also informative for the training process. In particular, we aim to generate synthetic samples belonging to the intersection of different class distributions known as "disagreement region" (Settles, 2012). These generated instances are informative for the training process since the learning model is uncertain about them (Houlsby et al., 2011).

3. "Information-Preserving" Data Augmentation for Active Learning

3.1. Bayesian Active Learning by Disagreement (BALD)

Let us denote the initial labeled data by $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ is the data sample labeled with $\mathbf{y}_i \in \mathcal{C} = \{1, 2, \ldots, C\}$ (C = # classes). By using Bayesian deep learning framework, we can obtain an estimate of the posterior of the parameters θ of the model $\mathcal{M}$ given $\mathcal{D}$, namely $p(\theta|\mathcal{D})$. In Bayesian Active Learning by Disagreement (BALD) scheme (Houlsby et al., 2011), the most informative sample $\mathbf{x}^*$ is selected from the (unlabeled) pool data $\mathcal{D}_{\text{pool}}$ by (Houlsby et al., 2011):

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{D}_{\text{pool}}} a(\mathbf{x}, \mathcal{M})$$
$$= \arg\max_{\mathbf{x} \in \mathcal{D}_{\text{pool}}} H[\mathbf{y}|\mathbf{x}, \mathcal{D}] - \mathbb{E}_{\theta \sim p(\theta|\mathcal{D})}[H[\mathbf{y}|\mathbf{x}, \theta]], \quad (1)$$

where $a(\mathbf{x}, \mathcal{M})$ is the acquisition function, $H[\mathbf{y}|\mathbf{x}, \mathcal{D}]$ and $H[\mathbf{y}|\mathbf{x}, \theta]$ are represented by the Shannon entropy (Shannon, 2001) of the prediction $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$ and the distribution $p(\mathbf{y}|\mathbf{x}, \theta)$, respectively. The sample $\mathbf{x}^*$ is labeled with $\mathbf{y}^*$ (by an oracle), and the labeled data set is updated for the next training iteration: $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{x}^*, \mathbf{y}^*)$. That active selection framework is repeated until convergence.

In order to estimate the acquisition function in (1), Gal et al. (2017) introduced the Monte Carlo (MC) dropout method. This objective function can be approximated by its sample mean (Gal et al., 2017):

$$a(\mathbf{x}, \mathcal{M}) \approx -\sum_c \left(\frac{1}{T}\sum_t \hat{p}_c^t\right) \log\left(\frac{1}{T}\sum_t \hat{p}_c^t\right) + \frac{1}{T}\sum_{c,t} \hat{p}_c^t \log \hat{p}_c^t, \quad (2)$$

where T is the number of dropout iterations, $\hat{\mathbf{p}}^t = [\hat{p}_1^t, \ldots, \hat{p}_C^t] = \text{softmax}(f(\mathbf{x}; \theta^t))$, with f representing the network function parameterized by θ^t that is sampled from an estimate of the (commonly intractable) posterior $p(\theta|\mathcal{D})$ at the t-th iteration.

3.2. Generative Model and Bayesian Data Augmentation

In the iterative Bayesian data augmentation (BDA) framework (Tran et al., 2017), each iteration consists of two steps: synthetic data generation and model training. The BDA model comprises a generator (that generates new training samples from a latent variable), a discriminator (that discriminates between real and fake samples) and a classifier (that classifies the samples into one of the classes in $\mathcal{C}$). At the first step, given a latent variable $\mathbf{u}$ (e.g., a multivariate Gaussian variable) and a class label $\mathbf{y} \in \mathcal{C}$, the generator represented by a function $g(.)$ maps the tuple $(\mathbf{u}, \mathbf{y})$ to a data point $\mathbf{x}^a = g(\mathbf{u}, \mathbf{y}) \in \mathcal{X}$, and $(\mathbf{x}^a, \mathbf{y})$ is then added to $\mathcal{D}$ for model training. In (Tran et al., 2017), the authors also showed a weak convergence proof that is related to the improvement of the posterior distribution $p(\theta|\mathcal{D})$.

3.3. Bayesian Generative Active Deep Learning

The main technical contribution of this paper consists of combining BALD and BDA for generating new labeled samples that are informative for the training process (see Fig. 3).

We modify BDA (Tran et al., 2017) by conditioning the generation step on a sample $\mathbf{x}$ and a label $\mathbf{y}$ (instead of the latent variable $\mathbf{u}$ and label $\mathbf{y}$ in BDA). More specifically, the most informative sample $\mathbf{x}^*$ selected by solving (1) using the estimation (2) is pushed to go through a variational autoencoder (VAE) (Kingma & Welling, 2013), which contains an encoder $e(.)$ and a decoder $g(.)$, in order to generate the sample $\mathbf{x}'$, as follows:

$$\mathbf{x}' = g(e(\mathbf{x}^*)). \quad (3)$$

The training process of a VAE is performed by minimizing the "reconstruction loss" $\ell(\mathbf{x}^*, g(e(\mathbf{x}^*))$ (Kingma & Welling, 2013), where if the number of training iterations is sufficiently large, we have:

$$\|\mathbf{x}' - \mathbf{x}^*\| < \varepsilon, \quad (4)$$

with ε representing an arbitrarily small positive constant – see Fig. 4 for an evidence for that claim.
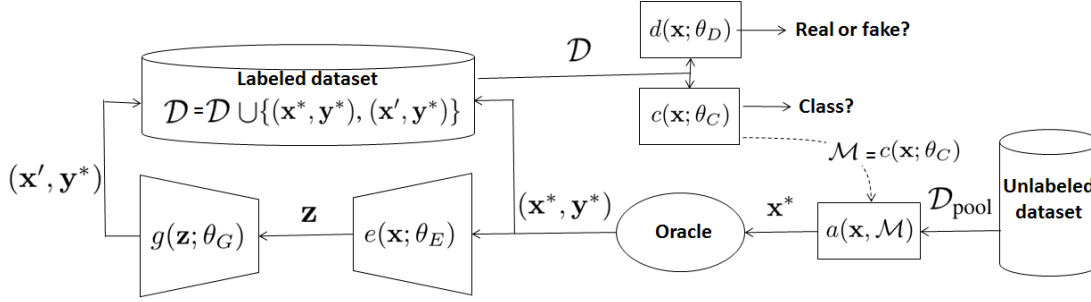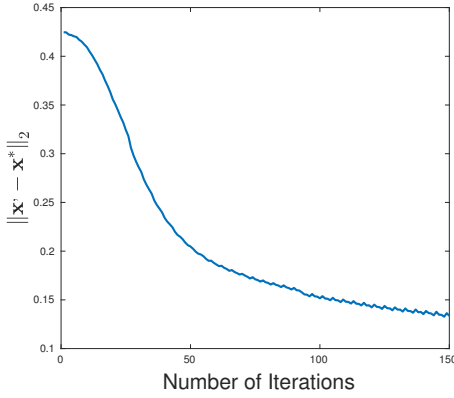
Figure 3. Network architecture of our proposed model.

Figure 4. Reduction of $\|\mathbf{x}' - \mathbf{x}^*\|$ as the training of the VAE model progresses (on CIFAR-100 using ResNet-18).

The label of $\mathbf{x}'$ is assumed to be $\mathbf{y}^*$ (i.e., the oracle's label for $\mathbf{x}^*$) and the current labeled data set is then augmented with $(\mathbf{x}^*, \mathbf{y}^*)$ and $(\mathbf{x}', \mathbf{y}^*)$, which are used for the next training iteration. To evaluate the "information content" of the generated sample $\mathbf{x}'$, which is measured by the value of the acquisition function at that point, $a(\mathbf{x}', \mathcal{M})$, we consider the following proposition.

Proposition 1. Assuming that there exists the gradient of the acquisition function $a(\mathbf{x}, \mathcal{M})$ with respect to the variable $\mathbf{x}$, namely $\nabla_x a(\mathbf{x}, \mathcal{M})$, and that $\mathbf{x}^*$ is an interior point of $\mathcal{D}_{\text{pool}}$, then $a(\mathbf{x}', \mathcal{M}) \approx a(\mathbf{x}^*, \mathcal{M})$ (i.e., the absolute difference between these values are within a certain range). Consequently, the sample $\mathbf{x}'$ generated from the most informative sample $\mathbf{x}^*$ by (3) is also informative.

Proof. Given the assumptions of Proposition 1, and due to the fact that $\mathbf{x}^*$ is a local maximum of function $a(\mathbf{x}, \mathcal{M})$ (1), then $\mathbf{x}^*$ is a critical point of $a(\mathbf{x}, \mathcal{M})$, i.e.,

$$\nabla_x a(\mathbf{x}^*, \mathcal{M}) = \mathbf{0}. \qquad (5)$$

Condition (4), which is empirically verified by Fig. 4, indicates that $\mathbf{x}'$ belongs to a sufficiently small neighborhood of $\mathbf{x}^*$. Therefore, by using the first order Taylor approximation

of the function $a(\mathbf{x}', \mathcal{M})$ at the point $\mathbf{x}^*$ and (5), we obtain

$$
\begin{aligned}
a(\mathbf{x}', \mathcal{M}) &\approx a(\mathbf{x}^*, \mathcal{M}) + \nabla_x a(\mathbf{x}^*, \mathcal{M})^T (\mathbf{x}' - \mathbf{x}^*) \\
&\approx a(\mathbf{x}^*, \mathcal{M}), \qquad (6)
\end{aligned}
$$

where T denotes the transpose operator. Thus, the synthetic sample $\mathbf{x}'$ can also be considered informative. $\qquad \square$

4. Implementation

Our network, depicted in Fig. 3, comprises four components: a classifier $c(\mathbf{x}; \theta_C)$, an encoder $e(\mathbf{x}; \theta_E)$, a decoder/generator $g(\mathbf{z}; \theta_G)$ and a discriminator $d(\mathbf{x}; \theta_D)$. The classifier $c(.)$ can be represented by any modern deep convolutional neural network classifier (LeCun et al., 1998; He et al., 2016a;b), making our model quite flexible in the sense that we can use the top-performing classifier available in the field. Also, the generative part of the model is based on ACGAN (Odena et al., 2017) and VAE-GAN (Larsen et al., 2016), where the VAE decoder is also the generator of the GAN model – our model is referred to as VAE-ACGAN.

The VAE-GAN loss function (Larsen et al., 2016; Zhang et al., 2017) was formed by adding the reconstruction error in VAE to the GAN loss in order to penalize both *unrealisticness* and *mode collapse* in GAN training. Following that, the VAE-ACGAN loss of our proposed model is defined by

$$\mathcal{L} = \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{ACGAN}}, \qquad (7)$$

with the VAE loss (Kingma & Welling, 2013; Larsen et al., 2016) represented by a combination of the reconstruction loss $\mathcal{L}_{\text{rec}}$ and the regularization prior $\mathcal{L}_{\text{prior}}$, i.e.,

$$
\begin{aligned}
\mathcal{L}_{\text{VAE}} &= \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{prior}} \\
&= \mathcal{L}_{\text{rec}}(\mathbf{x}, g(e(\mathbf{x}; \theta_E); \theta_G) + D_{\text{KL}}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})), \quad (8)
\end{aligned}
$$

where $\mathbf{z} = e(\mathbf{x}; \theta_E) = q(\mathbf{z}|\mathbf{x})$, $p(\mathbf{z})$ is the prior distribution of $\mathbf{z}$ (e.g., $\mathcal{N}(\mathbf{0}, \mathbf{I})$) and $D_{\text{KL}}(q\|p) = \int q \log \frac{p}{q}$ denotes the Kullback-Leibler divergence operator. The ACGAN

Algorithm 1 Bayesian Generative Active Learning

Initialize network parameters $\theta_E, \theta_G, \theta_C, \theta_D$, and pre-train the classifier $c(\mathbf{x}; \theta_C)$ with $\mathcal{D}$

repeat

Pick the most informative $\mathbf{x}^*$ from $\mathcal{D}_{\text{pool}}$ with $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{D}_{\text{pool}}} a(\mathbf{x}, \mathcal{M})$ in (1) and (2), where $\mathcal{M}$ is represented by the classifier $c(\mathbf{x}; \theta_C)$;

Request the oracle to label the selected sample, which forms $(\mathbf{x}^*, \mathbf{y}^*)$

$\mathbf{z} \leftarrow e(\mathbf{x}^*; \theta_E)$

$\mathcal{L}_{\text{prior}} \leftarrow D_{\text{KL}}(q(\mathbf{z}|\mathbf{x}^*)\|p(\mathbf{z}))$

$\mathbf{x}' = g(e(\mathbf{x}^*); \theta_G)$

$\mathcal{L}_{\text{rec}} \leftarrow \mathcal{L}_{\text{rec}}(\mathbf{x}^*, \mathbf{x}')$

Sample $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$\mathcal{L}_{\text{ACGAN}} \leftarrow \log(d(\mathbf{x}^*)) + \log(1 - d(\mathbf{x}')) + \log(1 - d(g(\mathbf{u}))) + \log(\text{softmax}(c(\mathbf{x}^*))) + \log(\text{softmax}(c(\mathbf{x}'))) + \log(\text{softmax}(c(g(\mathbf{u}))))$

$\theta_E \leftarrow \theta_E - \nabla_{\theta_E}(\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{prior}})$

$\theta_G \leftarrow \theta_G - \nabla_{\theta_G}(\gamma\mathcal{L}_{\text{rec}} - \mathcal{L}_{\text{ACGAN}})$ (parameter $\gamma = 0.75$ (Larsen et al., 2016) in our experiments)

$\theta_D \leftarrow \theta_D - \nabla_{\theta_D}\mathcal{L}_{\text{ACGAN}}$

$\theta_C \leftarrow \theta_C - \nabla_{\theta_C}\mathcal{L}_{\text{ACGAN}}$

until convergence

loss (Odena et al., 2017) in (7) is computed by

$$
\begin{aligned}
\mathcal{L}_{\text{ACGAN}} = {}& \log(d(\mathbf{x}; \theta_D)) + \log(1 - d(g(\mathbf{z}; \theta_G); \theta_D)) \\
& + \log(1 - d(g(\mathbf{u}; \theta_G); \theta_D)) + \log(\text{softmax}(c(\mathbf{x}; \theta_C))) \\
& + \log(\text{softmax}(c(g(\mathbf{z}; \theta_G); \theta_C))) \\
& + \log(\text{softmax}(c(g(\mathbf{u}; \theta_G); \theta_C))), \quad (9)
\end{aligned}
$$

where $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The training process of the VAE-ACGAN network is presented in Algorithm 1.

5. Experiments and Results

In this section, we assess quantitatively our proposed Bayesian Generative Active Deep Learning in terms of classification performance measured by the top-1 accuracy [1]. In particular, our proposed algorithm, active learning using "information-preserving" data augmentation (AL w. VAEACGAN) is compared with active learning using BDA (AL w. ACGAN), BALD without using data augmentation (AL without DA), BDA without active learning (BDA) (Tran et al., 2017) (using full and partial training sets), and random selection as a function of the number of acquisition iterations and the percentage of training samples. Our experiments are performed on MNIST (LeCun et al., 1998), CIFAR-10, CIFAR-100 (Krizhevsky et al., 2012), and SVHN (Netzer et al., 2011). MNIST (LeCun et al., 1998) contains handwritten digits, (with $60,000$ training and $10,000$ testing sam-

---

[1]code available at `https://github.com/toantm/BGADL`

ples, and 10 classes), CIFAR-10 (Krizhevsky et al., 2012) is composed of 32×32 color images (with $50,000$ training and $10,000$ testing samples, and 10 classes), CIFAR-100 (Krizhevsky et al., 2012) is similar to CIFAR-10, but with 100 classes, and SVHN (Netzer et al., 2011) contains 32×32 street view house numbers (with 73257 training samples and 26032 testing samples, and 10 classes).

Given that our approach can use any classifier, we test it with ResNet18 (He et al., 2016a) and ResNet18pa (He et al., 2016b), which have shown to produce competitive classification results in several tasks. The sample acquisition setup for each data set is: 1) the number of samples in the initial training set is $1,000$ for MNIST, $5,000$ for CIFAR-10, $15,000$ for CIFAR-100, and $10,000$ for SVHN (the initial data set percentage was empirically set – with values below these amounts, we could not make the training process converge; 2) the number of acquisition iterations is 150 (50 for SVHN), where at each iteration 100 (500 for SVHN) samples are selected from $2,000$ randomly selected samples of the unlabeled data set $\mathcal{D}_{\text{pool}}$ (this fixed number of randomly selected samples almost certainly contains the most informative sample (He & Ma, 2013)). The training process was run with the following hyper-parameters: 1) the classifier $c(\mathbf{x}; \theta_C)$ used stochastic gradient descent with (lr=0.01, momentum=0.9); 2) the encoder $e(\mathbf{x}; \theta_E)$, generator $g(\mathbf{z}; \theta_G)$ and discriminator $d(\mathbf{x}; \theta_D)$ used Adam optimizer with (lr=0.0002, $\beta_1 = 0.5$, $\beta_2 = 0.999$); the mini-batch size is 100 for all cases.

Fig. 5 compares the classification performance of several methods as a function of the number of acquisition iterations and respective percentage of samples from the original training set used for modeling. The methods compared are: 1) BDA (Tran et al., 2017) modeled with the full training set (*BDA (full training)*) and $10\times$ data augmentation to be used as an upper bound for all other methods; 2) the proposed Bayesian generative active learning (*AL w. VAEACGAN*); 3) active learning using BDA (*AL w. ACGAN*); 4) BDA modeled with partial training sets (*BDA (partial training)*); 5) BALD (Gal et al., 2017; Houlsby et al., 2011) without data augmentation (*AL without DA*); and 6) random selection of training samples using the percentage of samples from the original training set (*Random selection*). Each point of the curves in Fig. 5 presents the result of one acquisition iteration, where each new point represents a growing percentage of the training set, as shown in the horizontal axis. In Fig. 5, *BDA (partial training)* relies on $2\times$ data augmentation, so it uses the same number of real and artificial training samples as *AL w. VAEACGAN* and *AL w. ACGAN* – this enables a fair comparison between these methods.

To show a more informative comparison of our proposed approach (*AL w. VAEACGAN*) with other methods presented in Fig. 5, especially with *AL w. ACGAN* and *BDA (partial*
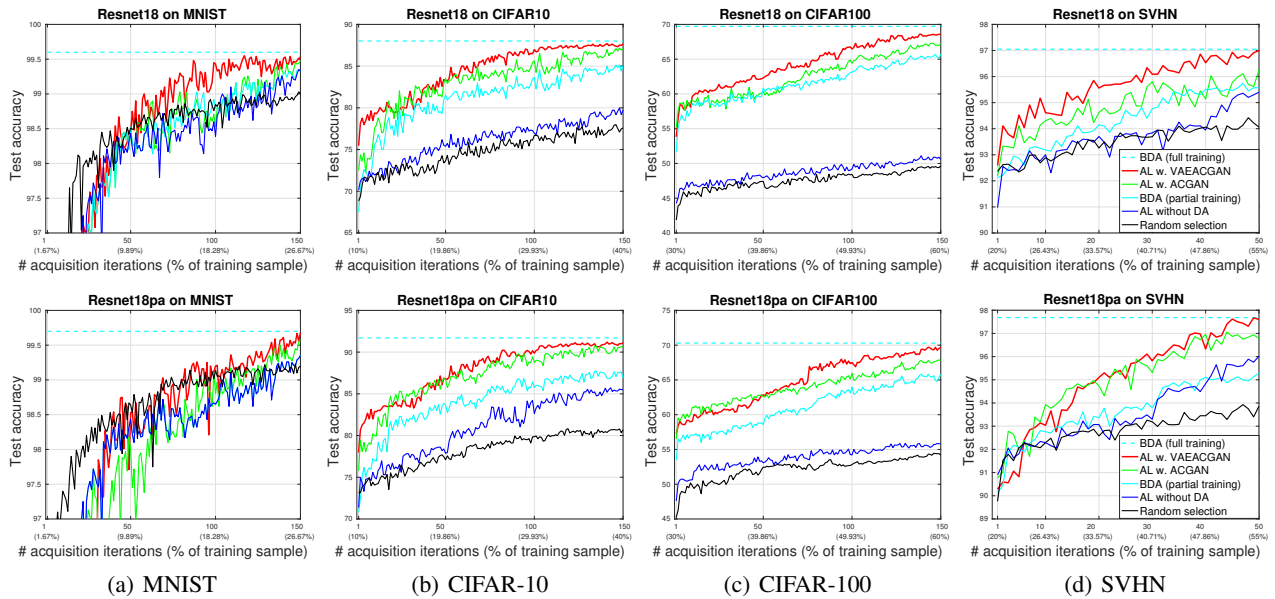
Figure 5. Training and classification performance of the proposed Bayesian generative active learning (*AL w. VAEACGAN*) compared to active learning using BDA (Tran et al., 2017) (*AL w. ACGAN*), BDA modeled with partial training sets (*BDA (partial training)*), BALD (Gal et al., 2017; Houlsby et al., 2011) without data augmentation (*AL without DA*), and random selection of training samples using the percentage of samples from the original training set (*Random selection*). The result for BDA modeled with the full training set (*BDA (full training)*) and $10\times$ data augmentation represents an upper bound for all other methods. This performance is measured as a function of the number of acquisition iterations and respective percentage of samples from the original training set used for modeling. First row shows these results using ResNet18 (He et al., 2016a), and second row shows ResNet18pa (He et al., 2016b) on MNIST (LeCun et al., 1998) (column 1), CIFAR-10 (column 2) CIFAR-100 (Krizhevsky et al., 2012) (column 3), and SVHN (Netzer et al., 2011) (column 4).

training), and active learning using PMDA (*AL w. PMDA*), using Resnet18 and Resnet18pa on MNIST, CIFAR-10, and CIFAR-100, we ran the experiments three times (with different random initialisations) and show the final classification results (mean $\pm$ stdev) in Tab. 1 (after 150 iterations).
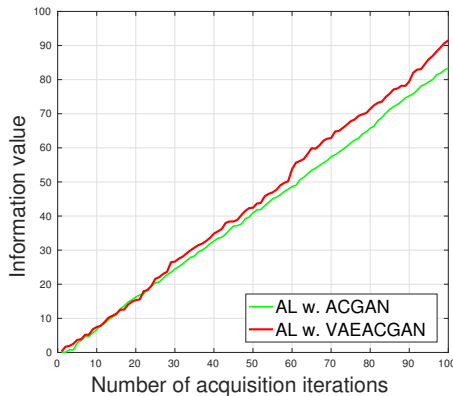

Figure 6. Average information value of samples measured by the acquisition function (2) of the samples generated by *AL w. ACGAN* and *AL w. VAEACGAN* using Resnet18 on CIFAR-100.

We also compare the average information value of samples measured by the acquisition function (2) of the samples generated by *AL w. ACGAN* and *AL w. VAEACGAN* in Fig. 6 using Resnet18 on CIFAR-100.
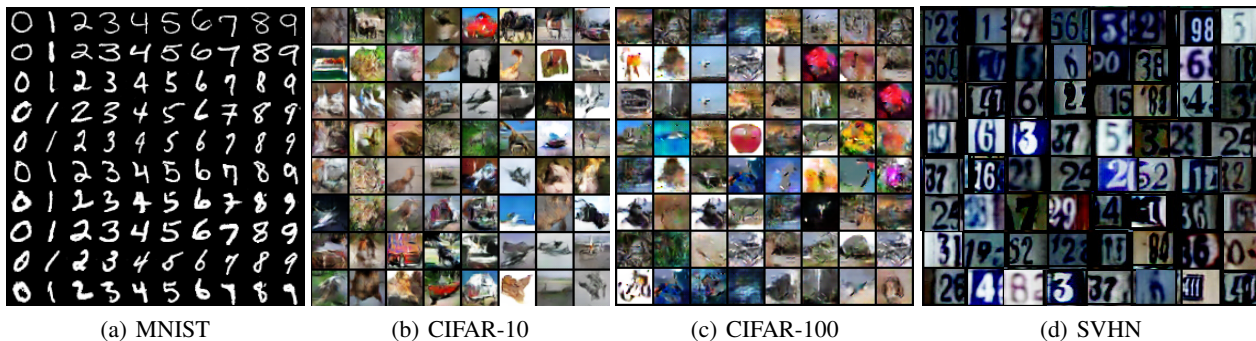
Figure 7 displays images generated by our generative model for each data set.

6. Discussion and Conclusions

Results in Fig. 5 consistently show (across different data sets and classification models) that our proposed Bayesian generative active learning (*AL w. VAEACGAN*) is superior to active learning with BDA (*AL w. ACGAN*), which is in fact an original model proposed by this paper. Even though informative samples are used for training *AL w. ACGAN*, the generated samples may not be informative, as depicted by Fig. 6 which shows that samples generated by *AL w. VAEACGAN* are more informative, particularly at latter stages of training. Nevertheless, the samples generated by *AL w. ACGAN* seem to be important for training given its better classification performance compared to *AL without DA*. Table 1 consistently shows that our proposed approach outperforms other methods on three data sets. In particular, the classification results by *AL w. VAEACGAN* are statistically significant with respect to *BDA (partial training)* on all those data sets, and with respect to *AL w. ACGAN* on CIFAR-{10, 100} for both models (i.e., $p \leq .05$, two-sample t-test for ResNet18 and ResNet18pa). Fig. 5 also shows that with a fraction of the training set, we are able to achieve a classification performance that is comparable with BDA using $10\times$ data

Table 1. Mean $\pm$ standard deviation of the classification accuracy on MNIST, CIFAR-10, and CIFAR-100 after 150 iterations over 3 runs

	AL w. VAEACGAN	AL w. ACGAN	MNIST AL w. PMDA	AL WITHOUT DA	BDA (PARTIAL TRAINING)	RANDOM SELECTION
RESNET18	**99.53 $\pm$ 0.05**	99.45 $\pm$ 0.02	99.37 $\pm$ 0.15	99.33 $\pm$ 0.10	99.33 $\pm$ 0.04	99.00 $\pm$ 0.13
RESNET18PA	**99.68 $\pm$ 0.08**	99.57 $\pm$ 0.07	99.49 $\pm$ 0.09	99.35 $\pm$ 0.11	99.35 $\pm$ 0.07	99.20 $\pm$ 0.12
			CIFAR-10			
RESNET18	**87.63 $\pm$ 0.11**	86.80 $\pm$ 0.45	82.17 $\pm$ 0.35	79.72 $\pm$ 0.19	85.08 $\pm$ 0.31	77.29 $\pm$ 0.23
RESNET18PA	**91.13 $\pm$ 0.10**	90.70 $\pm$ 0.24	87.70 $\pm$ 0.39	85.51 $\pm$ 0.21	86.90 $\pm$ 0.27	80.69 $\pm$ 0.19
			CIFAR-100			
RESNET18	**68.05 $\pm$ 0.17**	66.50 $\pm$ 0.63	55.24 $\pm$ 0.57	50.57 $\pm$ 0.20	65.76 $\pm$ 0.40	49.67 $\pm$ 0.52
RESNET18PA	**69.69 $\pm$ 0.13**	67.79 $\pm$ 0.76	59.67 $\pm$ 0.60	55.82 $\pm$ 0.31	65.79 $\pm$ 0.51	54.77 $\pm$ 0.29

(a) MNIST   (b) CIFAR-10   (c) CIFAR-100   (d) SVHN

Figure 7. Images generated by our proposed *AL w. VAEACGAN* approach for each data set.

augmentation over the entire training set – this is evidence that the generation of informative training samples can use less human and computer resources for labeling the data set and training the model, respectively. When using MNIST and ResNet18, we let *AL w. VAEACGAN* run until it reaches a competitive accuracy with BDA, which happened at 150 iterations – this is then used as a stopping criterion for all methods. If we leave all models running for longer, both *AL w. ACGAN* and *AL w. VAEACGAN* converge to *BDA (full training)*, with *AL w. VAEACGAN* converging faster. Furthermore, results in Fig. 5 demonstrate that for training sets of similar sizes, our proposed *AL w. VAEACGAN* produces better classification results than *BDA (partial training)* for all experiments, re-enforcing the effectiveness of generating informative training samples. It can also be seen from Fig. 5 that, on MNIST, the active learning methods initially behave worse than random sampling, but after a certain number of training acquisition steps (around 75 steps and 13% of the training set), they start to produce better results. Although the main goal of this work is the proposal of a better training process, the quality of the images generated, shown in Fig. 7, is surprisingly high.

In this work we proposed a Bayesian generative active deep learning approach that consistently shows to be more effective than data augmentation and active learning in several classification problems. One possible weakness of our paper

is the lack of a comparison with the only other method in the literature that proposes a similar approach (Zhu & Bento, 2017). Although relevant to our approach, (Zhu & Bento, 2017) focuses on binary classification (that paper provides a brief discussion on the extension to multi-class, but does not show that extension explicitly), and the results shown in that paper are not competitive enough to be reported here. In principle, our proposed approach is model-agnostic, it therefore can be combined with several currently introduced active learning methods such as (Sener & Savarese, 2018; Ducoffe & Precioso, 2018; Gissin & Shalev-Shwartz, 2018). In the future, we plan to investigate how to generate samples directly using complex acquisition functions, such as the one in (2), instead of conditioning the sample generation on highly informative samples selected from the unlabeled data set. We also plan to work on the efficiency of our proposed method because its empirical computational cost is slightly higher than BDA (Tran et al., 2017) and BALD (Gal et al., 2017; Houlsby et al., 2011).

Acknowledgments

References

Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pp. 17–36, 2012.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. Active learning with statistical models. *Journal of artificial intelligence research*, 1996.

Ducoffe, M. and Precioso, F. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Gal, Y., Islam, R., and Ghahramani, Z. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pp. 1183–1192, 2017.

Gissin, D. and Shalev-Shwartz, S. Discriminative active learning. 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

He, H. and Ma, Y. *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons, 2013.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pp. 630–645. Springer, 2016b.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, pp. 3, 2017.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL http://dblp.uni-trier.de/db/journals/corr/corr1312.html#KingmaW13.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pp. 3581–3589, 2014.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., and Socher, R. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pp. 1378–1387, 2016.

Larsen, A. B. L., Snderby, S. K., Larochelle, H., and Winther, O. Autoencoding beyond pixels using a learned similarity metric. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/larsen16.html.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., and Sánchez, C. I. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

MacKay, D. J. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604, 1992.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 5, 2011.

Odena, A., Olah, C., and Shlens, J. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*, pp. 2642–2651, 2017.

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Liu, P. J., Liu, X., Sun, M., Sundberg, P., Yee, H., et al. Scalable and accurate deep learning for electronic health records. *arXiv preprint arXiv:1801.07860*, 2018.

Sener, O. and Savarese, S. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

Settles, B. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

Shannon, C. E. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

Simard, P. Y., Steinkraus, D., and Platt, J. C. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*, 2003.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 843–852. IEEE, 2017.

Tanner, M. A. *Tools for statistical inference*, volume 3. Springer, 1991.

Tran, T., Pham, T., Carneiro, G., Palmer, L., and Reid, I. A bayesian data augmentation approach for learning deep models. In *Advances in Neural Information Processing Systems*, pp. 2797–2806, 2017.

Yaeger, L., Lyon, R., and Webb, B. Effective training of a neural network character classifier for word recognition. In *NIPS*, volume 9, pp. 807–813, 1996.

Zhang, Z., Song, Y., and Qi, H. Gans powered by autoencoding a theoretic reasoning. In *ICML Workshop on Implicit Models*, 2017.

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