Bayesian Data Augmentation and Generative Active Learning for Robust Imbalanced Deep Learning

Toan Minh Tran

October, 2019

Thesis submitted for the degree of
Doctor of Philosophy
in
Computer Science
at The University of Adelaide
Faculty of Engineering, Computer and Mathematical Sciences
School of Computer Science

SUB CRUCE LUMEN

THE UNIVERSITY
of ADELAIDE

Abstract

Deep learning has become a leading machine learning approach in many domains such as image classification, face recognition, and autonomous driving cars. However, its success is predicated on the availability of immense labelled training sets. Furthermore, it is usually the case that these data sets need to be well-balanced, otherwise the performance of the trained model is compromised. The outstanding performance of deep learning compared to other traditional machine learning approaches is therefore traded off by the need of a significant amount of human resources for labelling and computational resources for training. Designing effective deep learning approaches that can perform well using small and imbalanced labelled training sets is essential since that will increase the use of deep learning in many real-life applications.

In this thesis, we investigate several learning approaches that aim to improve the data efficiency in training deep models. In particular, we propose novel effective learning methods that enable deep learning models to perform well with relatively small and imbalanced labelled training sets.

We first introduce a novel theoretically sound Bayesian data augmentation (BDA) method motivated by the fact that the current dominant data augmentation (DA), based on small geometric and appearance transformations of the original training samples, does not guarantee the usefulness and the realism of the generated samples. We formulate BDA with the generalised Monte-Carlo expectation maximisation (GMCEM). We theoretically show the weak convergence of GMCEM and introduce an implementation of BDA based on a variant of the generative adversarial network (GAN). We empirically demonstrate that our proposed BDA performs better than the dominant DA above.

One of the drawbacks of BDA mentioned above is that the generation of

synthetic training samples is performed without considering their informativeness to the training process. Therefore, we next propose a new Bayesian generative active deep learning (BGADL) approach that aims to train a generative model to produce novel informative training samples. We formulate this algorithm based on a theoretically sound combination of the Bayesian active learning by disagreement (BALD) and BDA, where BALD guides BDA to produce synthetic samples. We provide a formal proof that these generated samples are informative for the training process. We provide empirical evidence that our proposed BGADL outperforms BDA and BALD with respect to training efficiency and classification accuracy.

The Bayesian generative active deep learning above does not properly handle class imbalanced training that may occur in the updated training sets formed at each iteration of the algorithm. We extend BGADL with an approach that is robust to imbalanced training data by combining it with a sample re-weighting learning approach. We empirically demonstrate that the extended BGADL performs well on several imbalanced data sets and produce better classification results compared to other baselines.

In summary, the contributions of this thesis are the introduction of the following novel methods: Bayesian data augmentation, Bayesian generative active deep learning, and a robust Bayesian generative active deep learning for imbalanced learning. All of those contributions are supported by theoretical justifications, empirical evidence and published or submitted papers.

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signed: .... Date: ....

Preface

This thesis was written at the School of Computer Science, The University of Adelaide. The main parts of the thesis are based on the following published/submitted papers in which I am the primary author and have contributed approximately 70% to each of them:

1. T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid. A bayesian data augmentation approach for learning deep models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2797–2806, 2017.
2. T. Tran, T.-T. Do, I. Reid, and G. Carneiro. Bayesian generative active deep learning. In *International Conference on Machine Learning (ICML)*, pages 6295–6304, 2019.
3. Toan Tran, Ian Reid, Gustavo Carneiro. Bayesian Generative Active Deep Learning Applied to Imbalanced Learning. Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2019.

Dedication

To my wife Huong, and my son Hieu.

Acknowledgements

Doing my Ph.D. at the University of Adelaide has been one of the most challenging but interesting experiences of my life. My research would have been impossible without the support, encouragement, and help from my supervisors, my sponsors, my family, my colleagues, and friends. I am thankful to all of them, to whom I owe my present situation.

I would like to thank my supervisors, Prof. Gustavo Carneiro and Prof. Ian Reid for their great instructions, advice and timely encouragement during my entire Ph.D. journey. Especially, I would like to express the deepest gratitude to my principal supervisor, Prof. Gustavo Carneiro for his fantastic guidance, advice, and patience. Each time we met to discuss my research, he always listened attentively and asked the right questions, suggesting new insights or teaching me new tricks which I could apply to address my problems. I am also so grateful for his open-mindedness, he often gave me maximum freedom to explore new methods and develop novel ideas yet still always willing to help whenever I needed. In short, I would not have been able to achieve a fraction of what I did in my Ph.D. without his great guidance and unconditional support.

I gratefully acknowledge the financial support by Vietnam International Education Development (VIED), the University of Adelaide, the Australian Centre for Robotic Vision (ACRV), and my supervisors in the last three years. Also, my sincere thanks to Ms. Thuy Mai for her wonderful assistance and advice for my trips to NIPS'17 and ICML'19.

I also would like to express my appreciation to Prof. Lyle Palmer, Dr. Trung Pham and Dr. Thanh-Toan Do for their great collaborations and valuable comments when writing and revising the NIPS'17 and ICML'19 papers. I also would like to thank Dr. Thanh-Toan Do for providing me an opportunity to participate

in his CVPR'19 paper. I am grateful to Prof. Dinh Phung for inviting me to visit and give a talk to his research group at Monash University. Many thanks to Dr. Trung Le for his hospitality and great advice about my future research and career during my visit to Monash.

I would like to thank my colleagues and friends in the School of Computer Science, in which I shared many wonderful experiences and moments. A special mention to Gabriel for the long-lasting chats about our research and tennis, also for the great instructions on how to use the phoenix server, and to Rafael for the kind help with the Ubuntu and Python installation. Many thanks to Hayden, Gabriel, Cuong, Gerard for their excellent help with the thesis proofreading. I also want to say thank to my friends Linh, Dinh, Vy, Khuong, Lanh, Dzung, Hoa, Hanh, Trung, and many others for their kind help along the way. Very special gratitude goes to my housemates Trang and Khue, as well as their family for taking care of me and treating me like a family member during the last two years. Thank you all once again, your friendship and encouragement made my Ph.D. journey much more colourful and enjoyable.

I would like to sincerely thank my mum and dad, and my parents-in-law. Words are powerless to express my gratitude to their unconditional support and encouragement. Finally, heartfelt thanks go to my wife Huong and my lovely son Hieu who always support and encourage me to pursue my passion.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Although deep learning has been shown to be a dominant machine learning approach in image classification [53], speech recognition [39], face recognition [75,83], autonomous driving cars [9, 44] and many other domains such as drug discovery [59, 100] and genomics [29, 54, 60, 77], its training process tends to be inefficient since it relies heavily on not only sufficiently large [92] but also well-balanced [1, 46, 68] labelled training sets. The outstanding performance of deep learning compared to other traditional machine learning approaches [60] such as support vector machine (SVM) [14] or random forests (RFs) [6] is, therefore, traded off by the need for immense amounts of human resources for labelling and computational resources for training. Designing effective deep learning methods that can generalise well using relatively small and imbalanced data sets is, therefore, essential for researchers and practitioners since it can lead to a reduction in the need for large training sets, computational resources, and balanced training sets. Among current learning approaches proposed to improve the data efficiency in training deep models such as few-shot learning [56, 69] or continual learning [80], we are particularly interested in the following three approaches: active learning [42, 85], data augmentation [53, 99], and hybrid methods that combine these two approaches [51, 98, 108]. To handle the class imbalance data issue, current solutions aim to re-balance the data set using, for example, re-sampling [8, 46], or sample re-weighting [78] methods.

One of the most effective learning approaches aiming to reduce the need for a large labelled training set is pool-based active learning [32, 42, 85]. This pool-based active learning is motivated by the fact that gathering a large amount of unlabelled data is relatively effortless while obtaining labelled data is much more challenging and expensive. A typical active learning framework relies on a small labelled training data set and a large unlabelled pool data set, where the learner is initially modelled with the small annotated data set, and it then actively selects the most informative samples from the unlabelled data set by maximising an acquisition function that is used to evaluate the information value of an unlabelled data point to the training process. These most informative samples are then requested to be labelled by an oracle – these newly annotated samples are then used in the next training iterations. Although pool-based active learning has been shown to achieve a good accuracy using significantly less labelled training data compared to other passive learning methods [26, 32, 98], that classical active learning approach cannot be directly applied for the estimation of deep models since at the early stages of the training process, the active learner tends to over-fit the small initial labelled training sets [98].

Alternatively, if the unlabelled data set is not available or challenging to be gathered, then one reasonable solution (to reduce the need for a large training set) is to synthetically generate new training samples to avoid over-fitting – that approach is known as data augmentation (DA) [53, 99, 101]. The goal of data augmentation is to enlarge the existing training set without manually labelling training samples, which is commonly known to be time-consuming, expensive, subjective and prone to mistakes [99]. The most common data augmentation method involves an application of several small-scale linear transformations such as random rotation, translation or colour perturbation to a real annotated sample in order to preserve its ground truth label [53]. We refer to that data augmentation approach as the "poor man's" data augmentation (PMDA) [95, 99] since it is performed only once, and prior to the training process. That PMDA has been employed as a heuristic regularisation scheme and shown some practical benefits in several computer vision tasks [53, 89, 104]. However, a common drawback of PMDA is that the usefulness of the generated training samples is not guaranteed due to the strong assumption about the label-preserving small-scale transformations [99]. Consequently, PMDA can produce unrealistic samples and may fail to generate a large range of realistic training samples [99].

One of the drawbacks of both active learning and data augmentation approaches mentioned above is that they are not designed to handle the imbalanced training problem that occurs when some majority classes contain considerably more training samples than other minority classes. In particular, the active sample selection in active learning and the synthetic data augmentation are performed without regarding how balanced the newly updated labelled training data sets are. Such imbalanced learning issue, which appears in many real-world applications, such as cancer and fraud detection [46,105], protein fold classification [110] and weld flaw classification [103], can make model classification less effective due to a poor prediction on minority classes [46]. Unfortunately, good classification performance on the minority classes is often of interest in a typical imbalanced learning problem [103]. To address the imbalanced learning issue, one reasonable solution is to re-balance the original imbalanced data set by random re-sampling approaches, such as under-sampling the majority class or over-sampling the minority class [46]. Alternatively, this issue can be addressed by re-weighting the samples based on cost-sensitive weighting [47,97] or average loss [2,57], or based on a novel robust meta-learning approach that learns to re-weight all the samples of the original training set by using a weighted loss [78].

In this thesis, we address the training issues of deep models mentioned above by developing several novel effective learning methods that enable deep learning models to perform well using not only relatively small, but also imbalanced labelled training data sets. These proposed methods are: 1) Bayesian data augmentation [99], 2) Bayesian generative active deep learning [98], and 3) a novel extension of the Bayesian generative active deep learning that is robust to class imbalanced data. The details about the contributions for each of these proposed approaches are discussed in the following section.

1.1 Thesis Contributions

In Chapter 3, we first introduce a novel theoretically sound data augmentation, referred to as Bayesian data augmentation (BDA), that aims to generate novel synthetic samples from the approximately learned data distribution of the existing observed samples [95,99]. The main contributions of BDA are as follows:

- We formulate BDA [99] by introducing a variant of the expectation maximisation (EM) algorithm [18], called generalised Monte-Carlo expectation maximisation (GMCEM). That BDA formulation was inspired by the classical data augmentation using latent variable method [95].

- We theoretically show the weak convergence of BDA training. This weak convergence is related to an improvement of the posterior of the model parameters given the observed data at each iteration of the GMCEM algorithm.

- We introduce an implementation of the BDA framework with an extension of the generative adversarial network (GAN) [30]–this is depicted in Fig. 1.1. We also provide a connection between the classical data augmentation using latent variable in statistical learning and modern deep generative models by theoretically showing that the loss function of the GAN-based model is linked to the objective function of the GMCEM method above.

- We provide empirical evidence that shows the better classification performance of BDA compared to the PMDA approach.

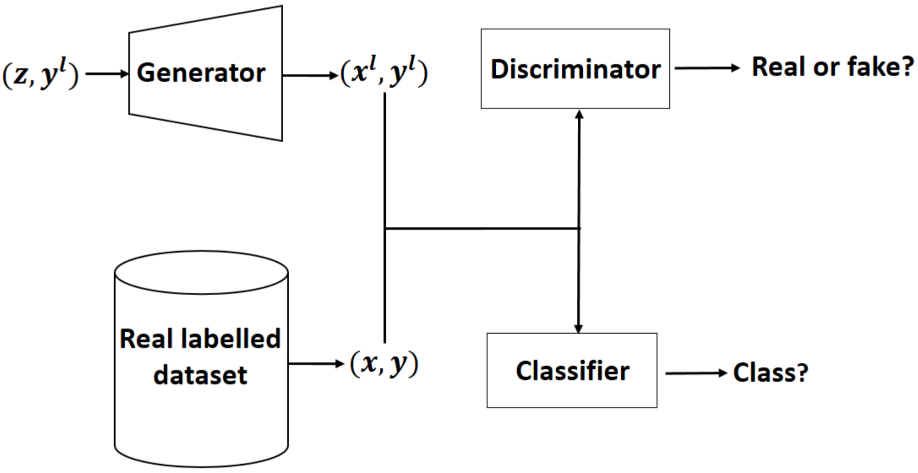


Figure 1.1: Bayesian data augmentation (BDA) [99], where $\mathbf{z}$ is a random noise, $\mathbf{y}^l$ is the class label for the generated sample, $(\mathbf{x}^l, \mathbf{y}^l)$ represents the generated sample, and $(\mathbf{x}, \mathbf{y})$ denotes the real labelled sample.

One key drawback of the BDA method proposed above is that it tends to waste not only training time but also computational resources due to the fact that the generation of new synthetic samples is performed without considering the informativeness (or usefulness) to the training process of the generated samples [98]. In an attempt to handle this issue, in Chapter 4, we propose the Bayesian generative active deep learning framework that aims to train a generative model to re-generate novel synthetic informative samples for the training process [98]. The contributions are the following:

- We provide a formulation of the Bayesian generative active deep learning algorithm based on a theoretically sound combination of the Bayesian active learning by disagreement (BALD) [26, 42] and BDA [99]. The proposed algorithm uses the active sample selection procedure in BALD to select the most informative samples from the unlabelled pool data – these informative samples are then used in the data augmentation scheme of BDA to generate new synthetic training samples–see Fig. 1.2.
- We theoretically prove that the generated samples are informative for the training process.
- We empirically demonstrate that our proposed Bayesian generative active deep learning [98] outperforms both BDA [99] and BALD [26, 42] in terms of training efficiency and classification performance.

One potential limitation of the Bayesian generative active deep learning [98] mentioned in Chapter 4 is that it does not handle the imbalanced training problem may occur in the newly updated labelled training data at each iteration of the Bayesian generative active deep learning [98], in Chapter 5, we extend that learning method to introduce a novel learning approach that is robust against class imbalance data. In our proposed method, the Bayesian generative active learning framework is combined with a recently proposed imbalanced learning approach [78]. In this chapter, we make the following contributions:

- We proposed the use the sample re-weighting method [78] to re-balance the newly updated labelled training data set at each active learning iteration in the Bayesian generative active deep learning approach [98]–this method is illustrated in Fig. 1.3.

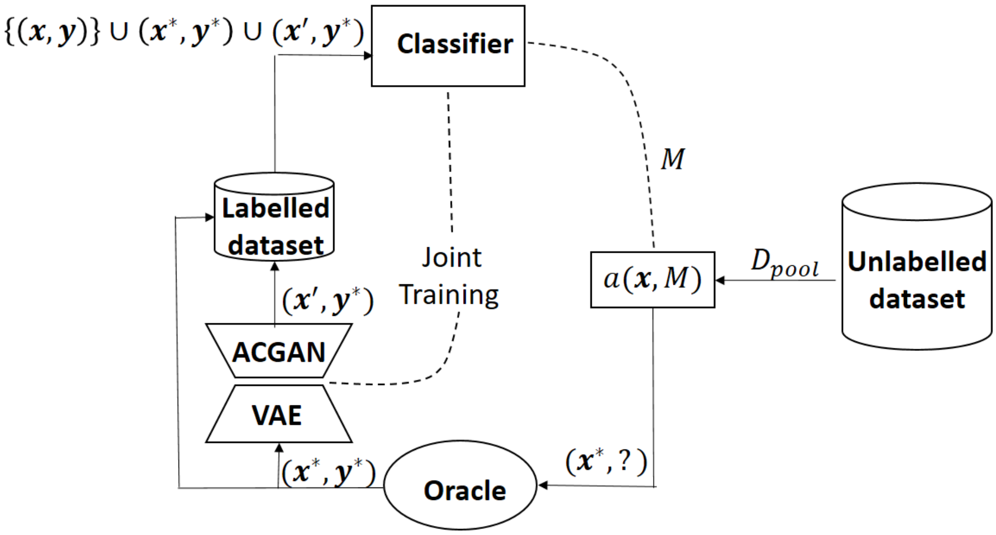


Figure 1.2: Bayesian generative active deep learning [98] framework, where $(\mathbf{x}, \mathbf{y})$ represents the initial labelled data, $(\mathbf{x}^*, \mathbf{y}^*)$ denotes the most informative sample selected by maximising the acquisition function $a(\mathbf{x}, M)$ over the unlabelled data D_{pool}, and $(\mathbf{x}^{'}, \mathbf{y}^*)$ is the generated sample that is theoretically shown to be informative. The details about VAE-ACGAN is mentioned in Chapter 2.

- We empirically show the considerable improvement with respect to the classification performance on several imbalanced data sets of our novel proposed method, compared to the original Bayesian generative active deep learning from Chapter 4, and other baselines.

1.2 Thesis Outline

The thesis contents is organised as follows:

- Chapter 2 provides literature review exploring the following relevant methods: 1) (Pool-based) active learning; 2) Bayesian active learning; 3) Data augmentation; 3) Generative active learning; and 4) Imbalanced learning.

- Chapter 3 focuses on the introduction of our proposed Bayesian data augmentation (BDA) that aims to train a generative model to automatically generate novel synthetic samples for the training of deep models.

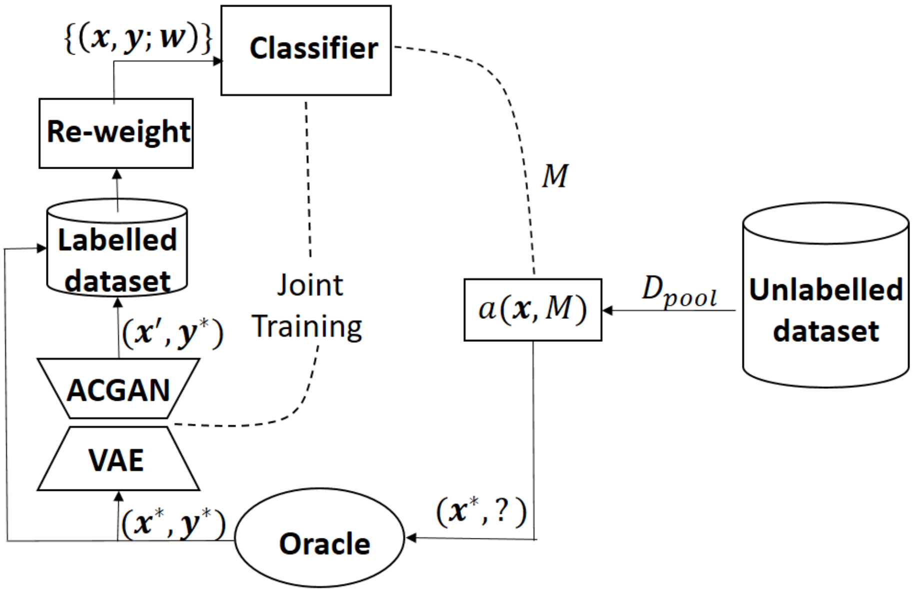


Figure 1.3: Our proposed Bayesian generative active deep learning that is robust to imbalanced data, where the sample re-weighting scheme [78] is used to re-balance the newly updated labelled data set at each iteration of the Bayesian generative active deep learning [98]. In this figure, $(\mathbf{x}, \mathbf{y}; \mathbf{w})$ represents the weighted sample, where $\mathbf{w}$ is the associated weight to the labelled data point $(\mathbf{x}, \mathbf{y})$.

- Chapter 4 introduces Bayesian generative active deep learning that targets the generation of informative data points for the training process.
- Chapter 5 first provides more comprehensive descriptions of BDA mentioned in chapter 3, and Bayesian generative active deep learning mentioned in chapter 4. This chapter also introduces our novel proposed robust Bayesian generative active deep learning to handle imbalanced data.
- Chapter 6 concludes the thesis, discuss the current limitations and future works for this research.

Chapter 2

Literature Review

In this chapter, we review current literature, and identify several limitations and research gaps in some of the relevant methods that we will address with our proposed approaches. We first introduce in Sec. 2.1 some background of the classical (pool-based) active learning framework, and the recently proposed Bayesian active learning by disagreement (BALD). In Sec. 2.2, we then analyse the dominant "poor man's" data augmentation (PMDA) that motivates our proposed theoretically sound Bayesian data augmentation method (BDA) [99]. We next discuss in Sec. 2.3 some generative active learning schemes, including our proposed Bayesian generative active deep learning [98]. In Sec. 2.4, we also explore generative adversarial network (GAN) and its variants employed in several corresponding demonstrations of our proposed algorithms mentioned above. Finally, we discuss in Sec. 2.5 several class imbalanced learning methods, followed by an introduction of our novel Bayesian generative active deep learning algorithm that is robust to imbalanced data.

2.1 Active Learning

Active learning is a sub-field of machine learning that aims to mitigate the "labelling bottleneck" of human annotation [85]–this labelling process is often time-consuming and prone to errors. This learning approach is well-motivated in many machine learning tasks such as speech recognition [109], information extraction [86], and classification and filtering [85], where unlabelled instances can be abundantly collected but labels are challenging and expensive to acquire. In active learning, the learner is allowed to access an unlabelled pool data set, and to request any samples from that pool to be labelled by an oracle (e.g., a human annotator). In this manner, the objective of active learning is to achieve high performance, while using less labelled training samples [85]. In contrast to the traditional passive learning, where the unlabelled samples are randomly chosen, in active learning, the learner can leverage its knowledge to request unlabelled instances to be labelled (by the oracle) and trained upon [32, 85] (see Fig. 2.1).

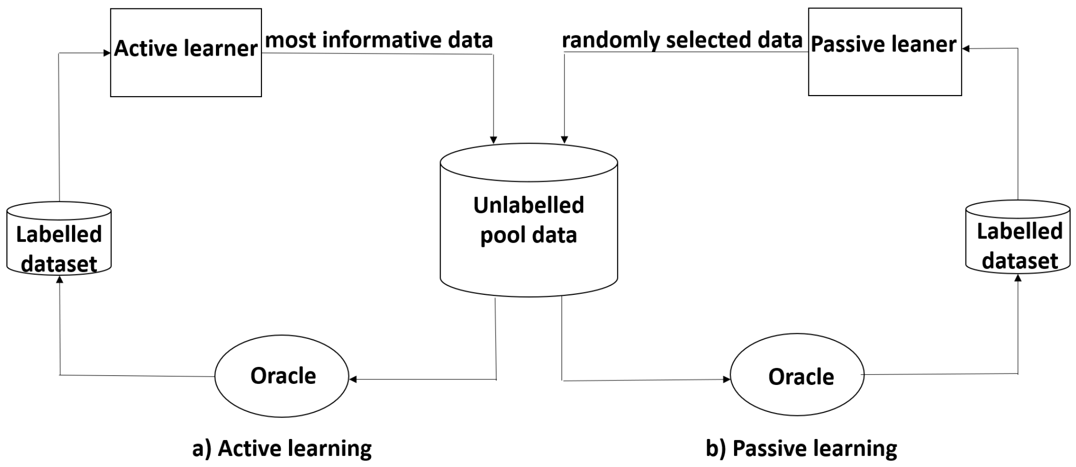


Figure 2.1: Comparison between a) active learning and b) passive learning [76, 85].

In principle, the setup of a typical (pool-based) active learning scheme consists of four main components: a learner (model), a small labelled training data set, a large unlabelled pool data set, and an acquisition function used to evaluate the information value of a sample for the training process. In that iterative active learning scheme [32, 85], having been initially modelled with the small labelled training data set, the active learner then automatically selects a subset of the most informative unlabelled samples by maximising the acquisition function over the pool data. These informative samples are annotated by an oracle, and then added to the original training data set for the next training iteration. In that active learning framework, the acquisition function can be estimated, for example, by the "expected informativeness" [67], or the (negative) "expected error" of the learner [11]. However, optimising these acquisition functions is challenging in the estimation of deep models since it requires the evaluation of the inverse of the Hessian matrix of the expected error with respect to the high-dimensional model parameter–that process is commonly known to be computationally challenging [98].

Houlsby et al. [42] proposed the Bayesian active learning by disagreement (BALD) scheme to facilitate the use of active learning in the estimation of deep models. The BALD algorithm is also known as the "information theoretic active learning" method, in which the acquisition function is estimated by the "mutual information" of the label of the sample with respect to the model parameters. In the BALD algorithm, this can be interpreted as the active learner aiming to select an unlabelled sample from the pool data such that the current model parameters (under the posterior distribution) firmly disagree about its outcome (label) [42]. Gal et al. [26] indicated that the BALD acquisition function estimation can be based on an approximation of the posterior distribution of the model parameters. Gal et al. [26] then introduced the use of a Monte-Carlo dropout method [25] to approximate the BALD acquisition function and some other types of acquisition functions that are usable in active deep learning. Recently, Kirsch et al. [50] introduced an extension of that BALD acquisition function, called BatchBALD, to target more diverse mini-batch of informative samples. Different from BALD, where each individual data point is acquired and then immediately used to re-train the model, in BatchBALD, a set of data points is selected by estimating the mutual information between the samples in the set and the model parameters [50].

2.2 Data Augmentation

The estimation of a deep learning model in active learning may lead to over-fitting since that model is assumed to initially rely on a small informative training set. One reasonable way to avoid that over-fitting issue is to enlarge the given labelled data set by generating novel synthetic training data points [53]. That learning approach is known as data augmentation, which has been widely employed in several computer vision tasks [53]. In the dominant "poor man's" data augmentation (PMDA) scheme [95,99], the generation of the new artificial training samples can be performed by applying several sufficiently small scale linear transformations [53, 89, 104] to the real training samples to preserve their labels–this is depicted in

Fig. 2.2.

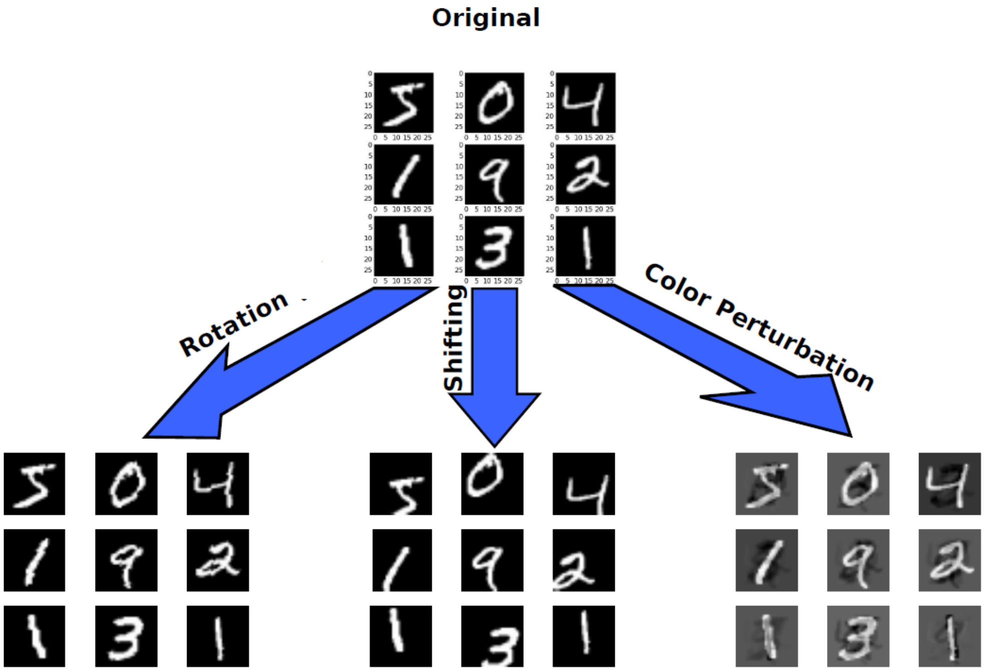


Figure 2.2: Synthetic images generated by several geometric and appearance transformations, including a rotation, translation, and colour perturbation from PMDA on MNIST [61].

One common problem in using PMDA is that the random linear transformations are often manually selected to target better performance of a given model with a specific data set–this task is known to be computationally expensive and often to require expertise [16]. Cubuk et al. [16] and Lim et al. [64] then strengthened PMDA by introducing the use of reinforcement learning [71,88,93], in which the reward function is defined as the validation accuracy on a target data set, to find an optimal subset of the given sets of the linear transformations. Although PMDA has been shown to work well in practice, it has not been properly tested. For example, it has not been clearly explained if the labels of the real samples are actually preserved through the small-scale linear transformations [99]. Moreover, in general, PMDA does not adapt well with the training process since the data augmentation procedure is executed only once, and prior to the training process [99].

Targeting a novel theoretically sound data augmentation, we proposed in Tran et al. [99] the Bayesian data augmentation (BDA) that aims to train a generative model to produce new synthetic samples belonging to an approximated data distribution of the real training samples. That BDA is inspired by the classical data augmentation using latent variables [94], in which the latent variables are used to accelerate the estimation of the posterior distributions of the model parameters given the observed data. An efficient way to present and then facilitate that DA using latent variables is based on the expectation maximisation (EM) algorithm [18]–this is depicted in Fig. 2.3. We provide a formulation of that BDA framework by adapting the (EM) algorithm to introduce its variant, called generalised Monte-Carlo expectation maximisation (GMCEM). More importantly, we theoretically prove the weak convergence of the GMCEM framework–this proof guarantees the improvement of the posterior distribution after each parameter estimation step. We also provide in [99] a demonstration of that BDA algorithm using a variant of generative adversarial network (GAN) [30], called ACGAN [74]. The details about our proposed BDA are mentioned in Chapter 3.

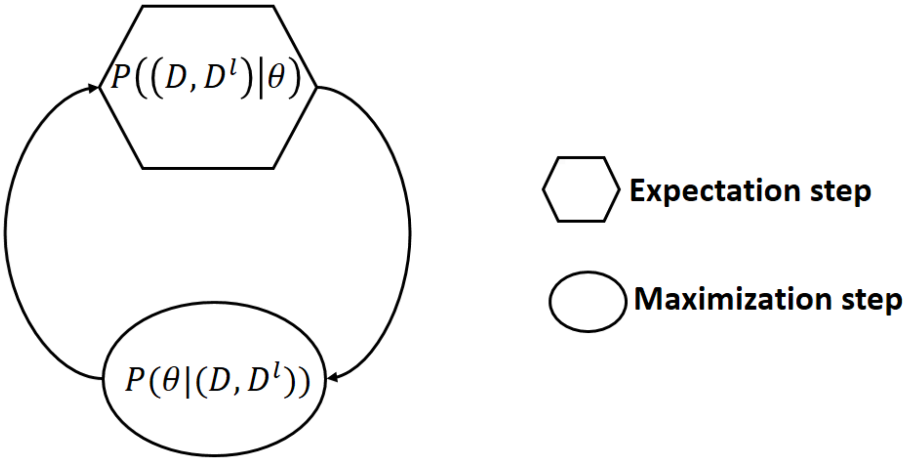


Figure 2.3: The expectation maximisation (EM) algorithm [96, 102], where D denotes the observed data set, D^l is the set of latent variables, which represents the generated samples in the BDA [99] model, and θ is the model parameters.

2.3 Generative Active Learning

The training performance of an active learning framework can be considerably accelerated by generating novel synthetic data points that are also informative for the training process–this method is known as generative active learning. The first generative active learning approach was proposed by Zhu and Bento [108], namely the generative adversarial active learning (GAAL) method that aims to generate novel synthetic training samples such that the current learner is uncertain about them. The learning principle of that GAAL method involves solving an optimisation problem, in which a pre-trained GAN model [30] is employed to generate novel informative samples [108]. Recently, Kong et al. [51] introduced the ActiveGAN algorithm that aims to directly generate labelled informative samples for the training of support vector machine (SVM) [14], without the need of an oracle. In that ActiveGAN, the estimation of the acquisition function (or the degree of uncertainty) is based on the distance from a sample to the hyper-plane of the SVM that was pre-trained with the existing labelled data points. This uncertainty will be integrated into the loss function for the training of an ACGAN model [74]. The generated informative samples from the trained ACGAN are then used to retrain the SVM classifier to achieve better performance. One common benefit of the GAAL and ActiveGAN methods is that it can generate informative data points for the training process without requiring the unlabelled pool data set, given that the GAN model is pre-trained and the optimisation problem can be solved efficiently. Nevertheless, directly applying those methods above in the estimation of deep models is challenging since they tend to rely on overly simple acquisition functions (e.g., the (negative) distance from the sample to the hyper-plane [85, 108])–such acquisition functions are shown not to be appropriate for active deep learning [26, 98].

Targeting a more effective generation process of informative samples for training deep models, we proposed in [98] a Bayesian generative active deep learning. This proposed approach consisted of a theoretically sound combination of Bayesian data augmentation (BDA) [99] and Bayesian active learning by disagreement (BALD) [26, 42]. In contrast to GAAL [108], which focus on the binary classification problems (a potential extension to multi-class problem was briefly discussed in [108] without any explicit execution), our proposed Bayesian generative active deep learning [98] is designed to handle multi-class problems using a deep classifier. Moreover, different from GAAL and ActiveGAN methods that involve 2-stage training, in which the generator and the classifier are independently trained, our proposed Bayesian generative active deep learning trains the learner and the generator jointly–this training principle allows them to "co-evolve" during the training process [98]. The details about the Bayesian generative active deep learning are given in Chapter 4.

2.4 Generative Adversarial Networks and Variational Auto-encoders

Generative adversarial network (GAN) [30] is one of the most effective deep generative model designed to learn how to produce novel synthetic data [29]. GAN has been widely used in creating texts, speech, videos, new art, and synthetic biology [15, 28]. In principle, GAN aims to estimate a generative model based on an "adversarial training process" performed by simultaneously training two deep learning models: a generator that learns to map a latent variable (e.g., random noise) to a "realistically looking" sample (i.e., the sample that belongs to a good approximation of the (unknown) ground truth data distribution), and a discriminator that estimates the probability that a sample came from the true data distribution rather than from the generator [28]. The performance of a GAN model is commonly evaluated by both the quality and the diversity of the generated samples that can be quantitatively measured by the inception score (IS) [82] and the Frechet inception score (FID) [38]. Among the extensions of GAN proposed to improve the synthetic image quality, one is particularly interesting: ACGAN [74], where the generator is conditioned on the label of the generated sample, and the discriminator is used to both identify the real/fake samples and classify that sample (see Fig. 2.4). The ACGAN model is adapted to demonstrate our proposed Bayesian data augmentation algorithm in [99].

One of the most challenging problems that can harm the ability of a GAN model to generate diverse data is "mode collapsing" or "mode missing", where the generator only focuses on generating synthetic samples from a few modes

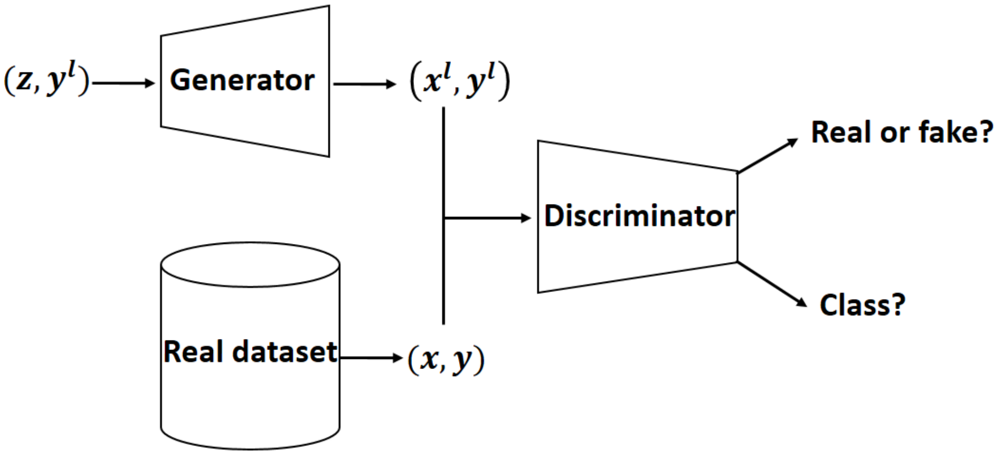


Figure 2.4: ACGAN model [74], where the generator is learned to map a tuple $(\mathbf{z}, \mathbf{y}^l)$, where $\mathbf{z}$ is a random noise, and $\mathbf{y}^l$ is a class label, to a synthetic labelled sample $(\mathbf{x}^l, \mathbf{y}^l)$. The discriminator is used not only to identify if a sample is real or fake but also to classify that sample.

instead of the whole data distribution [41,91]. In other words, that mode collapsing issue occurs when the generator learns to map several different latent variables values to the same output data point [28], thereby many real samples may have significantly small probability to be generated by the generator. One effective method that mitigates the "mode collapsing" issue in GAN training is to use variational autoencoder generative adversarial network (VAE-GAN) [58]–that is a combination of VAE [49, 79] and GAN, where these generative models are linked by the decoder/generator [107]. To provide an implementation for our proposed Bayesian generative active deep learning [98] that aims to generate a new synthetic training data point that is also informative for the training process, we modified that combined network to introduce the VAE-ACGAN model, where the generator is conditioned on both the selected informative sample and its label–this model is depicted in Fig. 2.5. The motivation of using that VAE-ACGAN in the Bayesian generative active deep learning [98] framework is based on the "reconstruction property" of the VAE training that transfers the information value from the selected sample to the novel generated data point–this is theoretically guaranteed in [98].

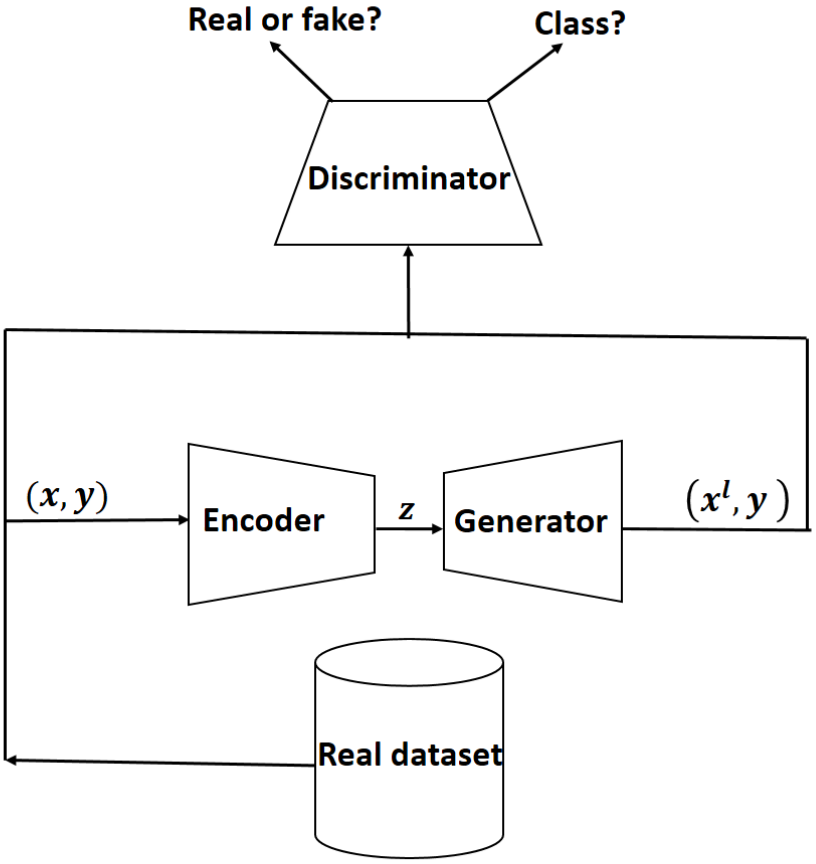


Figure 2.5: VAE-ACGAN model [58,98], where the encoder maps a labelled sample $(\mathbf{x}, \mathbf{y})$ to a latent variable $\mathbf{z}$. The generator/decoder is, therefore, conditioned on the sample $\mathbf{x}$ and its label $\mathbf{y}$ to generate a novel labelled sample $(\mathbf{x}^l, \mathbf{y})$. The discriminator estimates if a sample is real or fake, and classifies that sample.

2.5 Imbalanced Learning

One drawback of our Bayesian generative active deep learning [98] is that it does not handle the class imbalance issue that can arise in the newly updated labelled training set at each iteration. Designing effective learning methods using imbalanced data sets is essential since that can help improve prediction in the minority group. It is important to note that such minority group predictions can sometimes be more important than the prediction for the majority group (e.g., in medical diagnosis applications, diseases tend to occur in minority classes, and false negatives must be avoided) [46].

One reasonable solution to mitigate class imbalance is to modify the class distribution in the original imbalanced training data set. This method can be performed by under-sampling the majority class (i.e., remove random samples from majority group) or over-sampling the minority class (i.e., replicate random samples from minority group) [34, 46]. For example, in imbalanced active learning [34], Ertekin [22] introduced the virtual instance re-sampling technique using active learning (VIRTUAL), in which the generation of the novel synthetic sample is only performed on the selected informative samples that belong to the minority group. One limitation of that over-sampling VIRTUAL method is that it focuses only on the binary classification problem. Although these simple random re-sampling approaches can partly reduce the imbalance level in the training data, there are still several drawbacks that limit the ability to apply them in the field. In particular, under-sampling can reduce the target information value of the model, while over-sampling tends to increase the training time and computational resources, which can even lead to over-fitting" [8, 46]. Recently, Ren et al. [78] proposed a novel robust meta-learning method that aims to learn to re-weight the training samples by minimising the weighted loss function associated to the performance on a balanced validation set. Different from the random re-sampling techniques mentioned above, this sample re-weighting method aims to re-balance a skewed data set without changing its original size. We then combine this sample re-weighting method [78] with the Bayesian generative active deep learning [98] to propose a novel algorithm that is robust against class imbalance data. In particular, at each iteration of the Bayesian generative active deep learning framework [98], the sample re-weighting procedure is used to handle the class imbalance issue that may appear in the newly updated training data set.

CHAPTER 3

A Bayesian Data Augmentation Approach for Learning Deep Models

The work contained in this chapter has been published as the following paper:

T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid. A bayesian data augmentation approach for learning deep models. In *Advances in Neural Information Processing Systems*, pages 2797–2806, 2017 [99].

Statement of Authorship

Title of Paper	A Bayesian Data Augmentation Approach for Learning Deep Models
Publication Status	☒ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and Unsubmitted work written in manuscript style
Publication Details	T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid. A bayesian data augmentation approach for learning deep models. In Advances in Neural Information Processing Systems (NIPS), pages 2797–2806, 2017.

Principal Author

Name of Principal Author (Candidate)	Toan Minh Tran		
Contribution to the Paper	- Developed the idea of the paper - Implemented the proposed algorithm - Wrote and revised the manuscript		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	15/9/2019

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Trung Pham		
Contribution to the Paper	- Helped with the idea of the paper - Suggested some ideas to implement the proposed algorithm - Wrote and revised the manuscript		
Signature		Date	15-09-2019

Name of Co-Author	Gustavo Carneiro		
Contribution to the Paper	- Developed the idea of the paper - Suggested some ideas to implement the proposed algorithm - Supervised the development of the work - Wrote and revised the manuscript		
Signature		Date	15-09-2019

Name of Co-Author	Lyle Palmer		
Contribution to the Paper	- Helped with the idea of the paper		
Signature		Date	17-9-19

Name of Co-Author	Ian Reid		
Contribution to the Paper	- Helped with the idea and wirting process of the paper - Suggested some ideas to implement the proposed algorithm		
Signature		Date	19/9/19

Abstract

Data augmentation is an essential part of the training process applied to deep learning models. The motivation is that a robust training process for deep learning models depends on large annotated data sets, which are expensive to be acquired, stored and processed. Therefore a reasonable alternative is to be able to automatically generate new annotated training samples using a process known as data augmentation. The dominant data augmentation approach in the field assumes that new training samples can be obtained via random geometric or appearance transformations applied to annotated training samples, but this is a strong assumption because it is unclear if this is a reliable generative model for producing new training samples. In this paper, we provide a novel Bayesian formulation to data augmentation, where new annotated training points are treated as missing variables and generated based on the distribution learned from the training set. For learning, we introduce a theoretically sound algorithm — generalised Monte Carlo expectation maximisation, and demonstrate one possible implementation via an extension of the Generative Adversarial Network (GAN). Classification results on MNIST, CIFAR-10 and CIFAR-100 show the better performance of our proposed method compared to the current dominant data augmentation approach mentioned above — the results also show that our approach produces better classification results than similar GAN models.

3.1 Introduction

Deep learning has become the “backbone” of several state-of-the-art visual object classification [35,53,81,90], speech recognition [17,31,40], and natural language processing [12,13,106] systems. One of the many reasons that explains the success of deep learning models is that their large capacity allows for the modelling of complex, high dimensional data patterns. The large capacity allowed by deep learning is enabled by millions of parameters estimated within annotated training sets, where generalisation tends to improve with the size of these training sets. One way of acquiring large annotated training sets is via the manual (or “hand”) labelling of training samples by human experts — a difficult and sometimes subjective task that is expensive and prone to mistakes. Another way of producing such large training sets is to artificially enlarge existing training data sets — a process that is commonly known in computer science as data augmentation (DA).

In computer vision applications, DA has been predominantly developed with the application of simple geometric and appearance transformations on existing annotated training samples in order to generate new training samples, where the transformation parameters are sampled with additive Gaussian or uniform noise. For instance, for ImageNet classification [19], new training images can be generated by applying random rotations, translations or colour perturbations to the annotated images [53]. Such a DA process based on "label-preserving" transformations assumes that the noise model over these transformation spaces can represent with fidelity the processes that have produced the labelled images. This is a strong assumption that to the best of our knowledge has not been properly tested[1]. In fact, this commonly used DA process is known as "poor man's" data augmentation (PMDA) [95] in the statistical learning community because new synthetic samples are generated from a distribution estimated only once at the beginning of the training process.

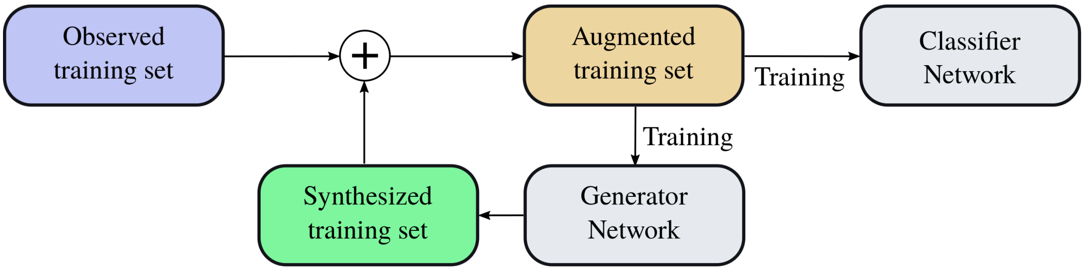


Figure 3.1: An overview of our Bayesian data augmentation algorithm for learning deep models. In this analytic framework, the generator and classifier networks are jointly learned, and the synthesised training set is continuously updated as the training progresses.

In the current manuscript, we propose a novel Bayesian DA approach for training deep learning models. In particular, we treat synthetic data points as instances of a random latent variable, which are drawn from a distribution learned

[1] It has not been clearly explained if the labels of the real samples are actually preserved through the small-scale linear transformations [99]

from the given annotated training set. Effectively, rather than generating new synthetic training data prior to the training process using pre-defined transformation spaces and noise models, our approach generates new training data as the training progresses using samples obtained from an iteratively learned training data distribution. Fig. 3.1 shows an overview of our proposed data augmentation algorithm.

The development of our approach is inspired by DA using latent variables proposed by the statistical learning community [96], where the motivation is to introduce latent variables to facilitate the computation of posterior distributions. However, directly applying this idea to deep learning is challenging because sampling millions of network parameters is computationally difficult. By replacing the estimation of the posterior distribution by the estimation of the maximum a posteriori (MAP) probability, one can employ the Expectation Maximization (EM) algorithm, if the maximisation of such augmented posteriors is feasible. Unfortunately, this is not the case for deep learning models, where the posterior maximisation cannot reliably produce a global optimum. An additional challenge for deep learning models is that it is nontrivial to compute the expected value of the network parameters given the current estimate of the network parameters and the augmented data.

In order to address such challenges, we propose a novel Bayesian DA algorithm, called Generalized Monte Carlo Expectation Maximization (GMCEM), which jointly augments the training data and optimises the network parameters. Our algorithm runs iteratively, where at each iteration we sample new synthetic training points and use Monte Carlo to estimate the expected value of the network parameters given the previous estimate. Then, the parameter values are updated with stochastic gradient decent (SGD). We show that the augmented learning loss function is actually equivalent to the expected value of the network parameters, and that therefore we can guarantee weak convergence. Moreover, our method depends on the definition of predictive distributions over the latent variables, but the design of such distributions is hard because they need to be sufficiently expressive to model high-dimensional data, such as images. We address this challenge by leveraging the recent advances reached by deep generative models [30], where data distributions are implicitly represented via deep neural networks whose parameters are learned from annotated data.

We demonstrate our Bayesian DA algorithm in the training of deep learning classification models [36,37]. Our proposed algorithm is realised by extending a generative adversarial network (GAN) model [30,70,74] with a data generation model and two discriminative models (one to discriminate between real and fake images and another to discriminate between the dataset classes). One important contribution of our approach is the fact that the modularity of our method allows us to test different models for the generative and discriminative models – in particular, we are able to test several recently proposed deep learning models [36, 37] for the dataset class classification. Experiments on MNIST, CIFAR-10 and CIFAR-100 datasets show the better classification performance of our proposed method compared to the current dominant DA approach.

3.2 Related Work

3.2.1 Data Augmentation

Data augmentation (DA) has become an essential step in training deep learning models, where the goal is to enlarge the training sets to avoid over-fitting. DA has also been explored by the statistical learning community [18,96] for calculating posterior distributions via the introduction of latent variables. Such DA techniques are useful in cases where the likelihood (or posterior) density functions are hard to maximise or sample, but the augmented density functions are easier to work. An important caveat is that in statistical learning, latent variables may not lie in the same space of the observed data, but in deep learning, the latent variables representing the synthesised training samples belong to the same space as the observed data.

Synthesising new training samples from the original training samples is a widely used DA method for training deep learning models [53,89,104]. The usual idea is to apply either additive Gaussian or uniform noise over pre-determined families of transformations to generate new synthetic training samples from the original annotated training samples. For example, Yaeger et al. [104] proposed the "stroke warping" technique for word recognition, which adds small changes in skew, rotation, and scaling into the original word images. Simard et al. [89]

used a related approach for visual document analysis. Similarly, Krizhevsky et al. [53] used horizontal reflections and colour perturbations for image classification. Hauberg et al. [33] proposed a manifold learning approach that is run once before the classifier training begins, where this manifold describes the geometric transformations present in the training set.

Nevertheless, the DA approaches presented above have several limitations. First, it is unclear how to generate diverse data samples. As pointed out by Fawzi et al. [24], the transformations should be "sufficiently small" so that the ground truth labels are preserved. In other words, these methods implicitly assume a small scale noise model over a pre-determined "transformation space" of the training samples. Such an assumption is likely too restrictive and has not been tested properly. Moreover, these DA mechanisms do not adapt with the progress of the learning process— instead, the augmented data are generated only once and prior to the training process. This is, in fact, analogous to the Poor Man's Data Augmentation (PMDA) [95] algorithm in statistical learning as it is non-iterative. In contrast, our Bayesian DA algorithm iteratively generates novel training samples as the training progresses, and the "generator" is adaptively learned. This is crucial because we do not make a noise model assumption over pre-determined transformation spaces to generate new synthetic training samples.

3.2.2 Deep Generative Models

Deep learning has been widely applied in training discriminative models with great success, but the progress in learning generative models has proven to be more difficult. One noteworthy work in training deep generative models is the Generative Adversarial Networks (GANs) proposed by Goodfellow et al. [30], which, once trained, can be used to sample synthetic images. A typical GANs consist of one generator and one discriminator, both represented by deep learning models. In "adversarial training", the generator and discriminator play a "two-player minimax game", in which the generator tries to fool the discriminator by rendering images as similar as possible to the real images, and the discriminator tries to distinguish the real and fake ones. Nonetheless, the synthetic images generated by GAN are of low quality when trained on the data sets with high variability [20]. Variants of GAN have been proposed to improve the quality of the synthetic images [10,70,73,74]. For instance, conditional GAN [70] improves the original GAN by making the generator conditioned on the class labels. Auxiliary classifier GAN (AC-GAN) [74] additionally forces the discriminator to classify both real-or-fake sources as well as the class labels of the input samples. These two works have shown significant improvement over the original GAN in generating photo-realistic images. So far these generative models mainly aim at generating samples of high-quality, high-resolution photo-realistic images. In contrast, we explore generative models (in the form of GANs) in our proposed Bayesian DA algorithm for improving classification models.

3.3 Data Augmentation Algorithm in Deep Learning

3.3.1 Bayesian Neural Networks

Our goal is to estimate the parameters of a deep learning model using an annotated training set denoted by $\mathcal{Y} = \{\mathbf{y}_n\}_{n=1}^{N}$, where $\mathbf{y} = (t, \mathbf{x})$, with annotations $t \in \{1, ..., K\}$ ($K =$ # Classes), and data samples represented by $\mathbf{x} \in \mathbb{R}^D$. Denoting the model parameters by θ, the training process is defined by the following optimisation problem:

$$\theta^* = \arg\max_{\theta} \log p(\theta|\mathbf{y}), \tag{3.1}$$

where the observed posterior $p(\theta|\mathbf{y}) = p(\theta|t, \mathbf{x}) \propto p(t|\mathbf{x}, \theta)p(\mathbf{x}|\theta)p(\theta)$.

Assuming that the data samples in $\mathcal{Y}$ are conditionally independent, the cost function that maximises (3.1) is defined as [5]:

$$\frac{1}{N} \log p(\theta|\mathbf{y}) \approx \log p(\theta) + \frac{1}{N} \sum_{n=1}^{N} (\log p(t_n|\mathbf{x}_n, \theta) + \log p(\mathbf{x}_n|\theta)), \tag{3.2}$$

where $p(\theta)$ denotes a prior on the distribution of the deep learning model parameters, $p(t_n|\mathbf{x}_n, \theta)$ represents the conditional likelihood of label t_n, and $p(\mathbf{x}_n|\theta)$ is the likelihood of the data $\mathbf{x}$.

In general, the training process to estimate the model parameters θ tends to over-fit the training set $\mathcal{Y}$ given the large dimensionality of θ and the fact that $\mathcal{Y}$ does not have a sufficiently large amount of training samples. One of the main approaches designed to circumvent this over-fitting issue is the automated generation of synthetic training samples — a process known as data augmentation (DA). In this work, we propose a novel Bayesian approach to augment the training set, targeting a more robust training process.

3.3.2 Data Augmentation using Latent Variable Methods

The DA principle is to increase the observed training data $\mathbf{y}$ using a latent variable $\mathbf{z}$ that represents the synthesised data, so that the augmented posterior $p(\theta|\mathbf{y},\mathbf{z})$ can be easily estimated [95], leading to a more robust estimation of $p(\theta|\mathbf{y})$. The latent variable is defined by $\mathbf{z} = (t^a, \mathbf{x}^a)$, where $\mathbf{x}^a \in \mathbb{R}^D$ refers to a synthesised data point, and $t^a \in \{1, ..., K\}$ denotes the associated label.

The most commonly chosen optimisation method in these types of training processes involving a latent variable is the expectation-maximisation (EM) algorithm [18]. In EM, let θ^i denote the estimated parameters of the model of $p(\theta|\mathbf{y})$ at iteration i, and $p(\mathbf{z}|\theta^i, \mathbf{y})$ represents the conditional predictive distribution of $\mathbf{z}$. Then, the E-step computes the expectation of $\log p(\theta|\mathbf{y},\mathbf{z})$ with respect to $p(\mathbf{z}|\theta^i,\mathbf{y})$, as follows:

$$Q(\theta,\theta^i) = \mathbb{E}_{p(\mathbf{z}|\theta^i,\mathbf{y})} \log p(\theta|\mathbf{y},\mathbf{z}) = \int_{\mathbf{z}} \log p(\theta|\mathbf{y},\mathbf{z}) p(\mathbf{z}|\theta^i,\mathbf{y}) d\mathbf{z}. \tag{3.3}$$

The parameter estimation at the next iteration, θ^{i+1}, is then obtained at the M-step by maximising the Q function:

$$\theta^{i+1} = \arg\max_{\theta} Q(\theta,\theta^i). \tag{3.4}$$

The algorithm iterates until $||\theta^{i+1} - \theta^i||$ is sufficiently small, and the optimal θ^* is selected from the last iteration. The EM algorithm guarantees that the sequence $\{\theta^i\}_{i=1,2,...}$ converges to a stationary point of $p(\theta|\mathbf{y})$ [18, 95], given that the expectation in (3.3) and the maximisation in (3.4) can be computed exactly. In the convergence proof [18, 95], it is assumed that θ^i converges to θ^* as the number of iterations i increases, then the proof consists of showing that θ^* is a critical point of $p(\theta|\mathbf{y})$.

However, in practice, either the E-step or M-step or both can be difficult to compute exactly, especially when working with deep learning models. In such cases, we need to rely on approximation methods. For instance, Monte Carlo sampling method can approximate the integration in (3.3) (the E-step). This technique is known as Monte Carlo EM (MCEM) algorithm [95]. Furthermore, when the estimation of the global maximiser of $Q(\theta, \theta^i)$ in (3.4) is difficult, Dempster et al. [18] proposed the Generalized EM (GEM) algorithm, which relaxes this requirement with the estimation of θ^{i+1}, where $Q(\theta^{i+1}, \theta^i) > Q(\theta^i, \theta^i)$. The GEM algorithm is proven to have weak convergence [95], by showing that $p(\theta^{i+1}|\mathbf{y}) > p(\theta^i|\mathbf{y})$, given that $Q(\theta^{i+1}, \theta^i) > Q(\theta^i, \theta^i)$.

3.3.3 Generalized Monte Carlo EM Algorithm

With the latent variable $\mathbf{z}$, the augmented posterior $p(\theta|\mathbf{y}, \mathbf{z})$ becomes:

$$p(\theta|\mathbf{y}, \mathbf{z}) = \frac{p(\mathbf{y}, \mathbf{z}, \theta)}{p(\mathbf{y}, \mathbf{z})} = \frac{p(\mathbf{z}|\mathbf{y}, \theta)p(\theta|\mathbf{y})p(\mathbf{y})}{p(\mathbf{z}|\mathbf{y})p(\mathbf{y})} = \frac{p(\mathbf{z}|\mathbf{y}, \theta)p(\theta|\mathbf{y})}{p(\mathbf{z}|\mathbf{y})}, \tag{3.5}$$

where the E-step is represented by the following Monte-Carlo estimation of $Q(\theta, \theta^i)$:

$$\begin{aligned} \hat{Q}(\theta, \theta^i) &= \frac{1}{M} \sum_{m=1}^{M} \log p(\theta|\mathbf{y}, \mathbf{z}_m) \\ &= \log p(\theta|\mathbf{y}) + \frac{1}{M} \sum_{m=1}^{M} (\log p(\mathbf{z}_m|\mathbf{y}, \theta) - \log p(\mathbf{z}_\mathbf{m}|\mathbf{y})), \end{aligned} \tag{3.6}$$

where $\mathbf{z}_m \sim p(\mathbf{z}|\mathbf{y}, \theta^i)$, for $m \in \{1, ..., M\}$. In (3.6), if the label t^a_m of the m^{th} synthesised sample $\mathbf{z}_\mathbf{m}$ is known, then $\mathbf{x}^a_m$ can be sampled from the distribution $p(\mathbf{x}^a_m|\theta, \mathbf{y}, t^a_m)$. Hence, the conditional distribution $p(\mathbf{z}|\mathbf{y}, \theta)$ can be decomposed as:

$$p(\mathbf{z}|\mathbf{y}, \theta) = p(t^a, \mathbf{x}^a|\mathbf{y}, \theta) = p(t^a|\mathbf{x}^a, \mathbf{y}, \theta)p(\mathbf{x}^a|\mathbf{y}, \theta), \tag{3.7}$$

where $(t^a, \mathbf{x}^a)$ are conditionally independent of $\mathbf{y}$ given that all the information from the training set $\mathbf{y}$ is summarized in θ — this means that $p(t^a|\mathbf{x}^a, \mathbf{y}, \theta) = p(t^a|\mathbf{x}^a, \theta)$, and $p(\mathbf{x}^a|\mathbf{y}, \theta) = p(\mathbf{x}^a|\theta)$.

The maximisation of $\hat{Q}(\theta, \theta^i)$ with respect to θ for the M-step is re-formulated by first removing all terms that are independent of θ, which allows us to reach the following derivation (making the same assumption as in (3.2)):

$$\hat{Q}(\theta, \theta^i) = \log p(\theta) + \frac{1}{N} \sum_{n=1}^{N} (\log p(t_n | \mathbf{x}_n, \theta) + \log p(\mathbf{x}_n | \theta)) + \frac{1}{M} \sum_{m=1}^{M} \log p(\mathbf{z}_m | \mathbf{y}, \theta) \tag{3.8}$$

$$= \log p(\theta) + \frac{1}{N} \sum_{n=1}^{N} (\log p(t_n | \mathbf{x}_n, \theta) + \log p(\mathbf{x}_n | \theta)) +$$
$$+ \frac{1}{M} \sum_{m=1}^{M} (\log p(t_m^a | \mathbf{x}_m^a, \theta) + \log p(\mathbf{x}_m^a | \theta)).$$

Given that there is no analytical solution for the optimisation in (3.8), we follow the same strategy employed in the GEM algorithm, where we estimate θ^{i+1} so that $\hat{Q}(\theta^{i+1}, \theta^i) > \hat{Q}(\theta^i, \theta^i)$.

As the function $\hat{Q}(\cdot, \theta^i)$ is differentiable, we can find such θ^{i+1} by running one step of gradient ascent. It can be seen that our proposed optimization consists of a marriage between MCEM and GEM algorithms, which we name: Generalized Monte Carlo EM (GMCEM). The weak convergence proof of GMCEM is provided by Lemma 1.

Lemma 1. *Assuming that θ^{i+1} is obtained by a gradient ascent step, i.e., $\hat{Q}(\theta^{i+1}, \theta^i) \geq \hat{Q}(\theta^i, \theta^i)$, which is guaranteed from* (3.8)*, then the weak convergence (i.e. $p(\theta^{i+1} | \mathbf{y}) \geq p(\theta^i | \mathbf{y})$) will be fulfilled.*

Proof. Given $\hat{Q}(\theta^{i+1}, \theta^i) > \hat{Q}(\theta^i, \theta^i)$, then by taking the expectation on both sides, that is $\mathbb{E}_{p(\mathbf{z}|\mathbf{y},\theta^i)}[\hat{Q}(\theta^{i+1}, \theta^i)] > \mathbb{E}_{p(\mathbf{z}|\mathbf{y},\theta^i)}[\hat{Q}(\theta^i, \theta^i)]$, we obtain $Q(\theta^{i+1}, \theta^i) > Q(\theta^i, \theta^i)$, which is the condition for $p(\theta^{i+1} | \mathbf{y}) > p(\theta^i | \mathbf{y})$ proven from [95]. □

So far, we have presented our Bayesian DA algorithm in a very general manner. The specific forms that the probability terms in (3.8) take in our implementation are presented in the next section.

3.4 Implementation

In general, our proposed DA algorithm can be implemented using any deep generative and classification models which have differentiable optimisation functions. This is in fact an important advantage that allows us to use the most sophisticated extant models available in the field for the implementation of our algorithm. In this section, we present a specific implementation of our approach using state-of-the-art discriminative and generative models.

3.4.1 Network Architecture

Our network architecture consists of two models: a classifier and a generator. For the classifier, modern deep convolutional neural networks [36, 37] can be used. For the generator, we select the *adversarial* generative networks (GAN) [30], which include a generative model (represented by a deconvolutional neural network) and an authenticator model (represented by a convolutional neural network). This authenticator component is mainly used for facilitating the *adversarial* training. As a result, our network consists of a classifier (C) with parameters θ_C, a generator (G) with parameters θ_G and an Authenticator (A) with parameters θ_A. Fig. 3.2 compares our network architecture with other variants of GAN recently proposed [30,70,74]. On the surface, our network appears similar to AC-GAN [74], where the only difference is the separation of the classifier network from the authenticator network. However, this crucial modularisation enables our DA algorithm to replace GANs by other generative models that may become available in the future; likewise, we can use the most sophisticated classification models for C. Furthermore, unlike our model, the classification subnetwork introduced in AC-GAN mainly aims for improving the quality of synthesised samples, rather than for classification tasks. Nonetheless, one can consider AC-GAN as one possible implementation of our DA algorithm. Finally, our proposed GAN model is similar to the recently proposed triplet GAN [63] [2], but it is important to emphasise that triplet GAN was proposed in order to improve the training procedure for GANs, while our model represents a particular realisation of the proposed Bayesian DA algorithm, which is the main contribution of this paper.

[2] The triplet GAN [63] was proposed in parallel to this NIPS submission.

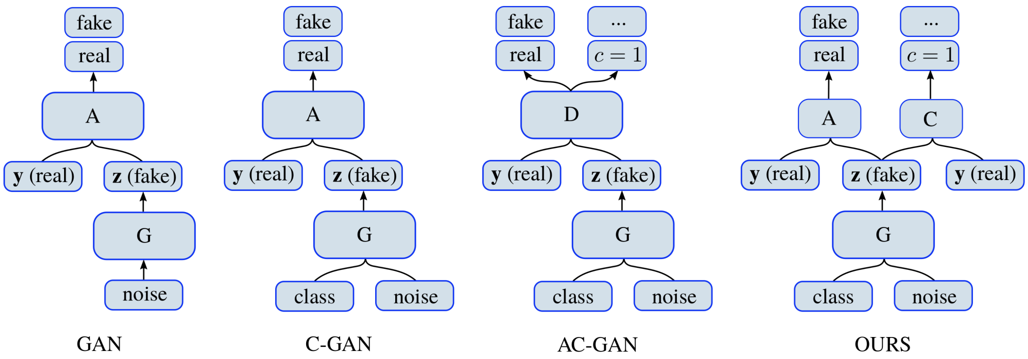


Figure 3.2: A comparison of different network architectures including GAN [30], C-GAN [70], AC-GAN [74] and ours. G: Generator, A: Authenticator, C: Classifier, D: Discriminator.

3.4.2 Optimization Function

Let us define $\mathbf{x} \in \mathbb{R}^D$, $\theta_C \in \mathbb{R}^C$, $\theta_A \in \mathbb{R}^A$, $\theta_G \in \mathbb{R}^G$, $u \in \mathbb{R}^\ell$, $c \in \{1, ..., K\}$, the classifier C, the authenticator A and the generator G are respectively defined by

$$f_C : \mathbb{R}^D \times \mathbb{R}^C \rightarrow [0,1]^K; \tag{3.9}$$

$$f_A : \mathbb{R}^D \times \mathbb{R}^A \rightarrow [0,1]^2; \tag{3.10}$$

$$f_G : \mathbb{R}^\ell \times \mathbb{Z}_+ \times \mathbb{R}^G \rightarrow \mathbb{R}^D. \tag{3.11}$$

The optimisation function used to train the classifier C is defined as:

$$J_C(\theta_C) = \frac{1}{N}\sum_{n=1}^{N} l_C(t_n|\mathbf{x}_n, \theta_C) + \frac{1}{M}\sum_{m=1}^{M} l_C(t_m^a|\mathbf{x}_m^a, \theta_C), \tag{3.12}$$

where $l_C(t_n|\mathbf{x}_n, \theta_C) = -\log\left(\text{softmax}(f_C(t_n = c; \mathbf{x}_n, \theta_C))\right)$.

The optimisation functions for the authenticator and generator networks are defined by [30]:

$$J_{AG}(\theta_A, \theta_G) = \frac{1}{N}\sum_{n=1}^{N} l_A(\mathbf{x}_n|\theta_A) + \frac{1}{M}\sum_{m=1}^{M} l_{AG}(\mathbf{x}_m^a|\theta_A, \theta_G), \tag{3.13}$$

where

$$l_A(\mathbf{x}_n|\theta_A) = -\log\left(\text{softmax}(f_A(input = real, \mathbf{x}_n, \theta_A)\right); \tag{3.14}$$

$$l_{AG}(\mathbf{x}_m^a|\theta_A, \theta_G) = -\log\left(1 - \text{softmax}(f_A(input = real, \mathbf{x}_m^a, \theta_G, \theta_A))\right). \tag{3.15}$$

Following the same training procedure used to train GANs [30,74], the optimisation is divided into two steps: the training of the discriminative part, consisting of minimising $J_C(\theta_C) + J_{AG}(\theta_A, \theta_G)$ and the training of the generative part consisting of minimising $J_C(\theta_C) - J_{AG}(\theta_A, \theta_G)$. This loss function can be linked to (3.8), as follows:

$$l_C(t_n|\mathbf{x}_n, \theta_C) = -\log p(t_n|\mathbf{x}_n, \theta), \tag{3.16}$$

$$l_C(t^a_m|\mathbf{x}^a_m, \theta_C) = -\log p(t^a_m|\mathbf{x}^a_m, \theta), \tag{3.17}$$

$$l_A(\mathbf{x}_n|\theta_A) = -\log p(\mathbf{x}_n|\theta), \tag{3.18}$$

$$l_{AG}(\mathbf{x}^a_m|\theta_A, \theta_G) = -\log p(\mathbf{x}^a_m|\theta). \tag{3.19}$$

3.4.3 Training

Training the network parameters θ follows the proposed GMCEM algorithm presented in Sec. 4.3. Accordingly, at each iteration we need to find θ^{i+1} so that $\hat{Q}(\theta^{i+1}, \theta^i) > \hat{Q}(\theta^i, \theta^i)$, which can be achieved using gradient decent. However, since the number of training and augmented samples (i.e., $N + M$) is large, evaluating the sum of the gradients over this whole set is computationally expensive. A similar issue was observed in contrastive divergence [7], where the computation of the approximate gradient required in theory an infinite number of Markov chain Monte Carlo (MCMC) cycles, but in practice, it was noted that only a few cycles were needed to provide a robust gradient approximation. Analogously, following the same principle, we propose to replace gradient decent by stochastic gradient decent (SGD), where the update from θ^i to θ^{i+1} is estimated using only a sub-set of the $M + N$ training samples. In practice, we divide the training set into batches, and the updated θ^{i+1} is obtained by running SGD through all batches (i.e, one epoch). We found that such strategy works well empirically, as shown in the experiments (Sec. 5.4).

3.5 Experiments

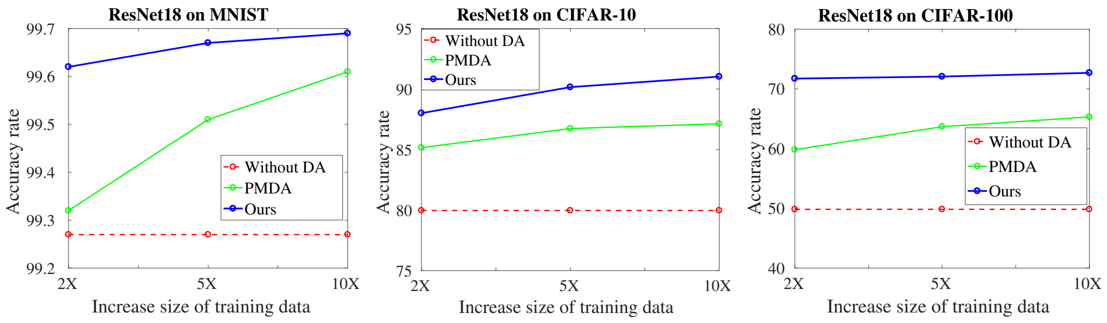
In this section, we compare our proposed Bayesian DA algorithm with the commonly used DA technique [53] (denoted as PMDA) on several image classification tasks (code available at: `https://github.com/toantm/keras-bda`). This comparison is based on experiments using the following three datasets: MNIST [61] (containing $60,000$ training and $10,000$ testing images of 10 handwritten digits), CIFAR-10 [52] (consisting of $50,000$ training and $10,000$ testing images of 10 visual classes like car, dog, cat, etc.), and CIFAR-100 [52] (containing the same amount of training and testing samples as CIFAR-10, but with 100 visual classes).

The experimental results are based on the top-1 classification accuracy as a function of the amount of data augmentation used – in particular, we try the following amounts of synthesised images M: a) $M = N$ (i.e., $2\times$ DA), $M = 4N$ ($5\times$ DA), and $M = 9N$ ($10\times$ DA). The PMDA is based on the use of a uniform noise model over a rotation range of $[-10, 10]$ degrees, and a translation range of at most 10% of the image width and height. Other transformations were tested, but these two provided the best results for PMDA on the data sets considered in this paper. We also include an experiment that does not use DA in order to illustrate the importance of DA in deep learning.

As mentioned in Sec. 5.1, one important contribution of our method is its ability to use arbitrary deep learning generative and classification models. For the generative model, we use the C-GAN [70] [3], and for the classification model we rely on the ResNet18 [36] and ResNetpa [37]. The architectures of the generator and authenticator networks, which are kept unchanged for all three datasets, can be found in the supplementary material. For training, we use Adadelta (with learning rate=1.0, decay rate=0.95 and epsilon=$1e-8$) for the Classifier (C), Adam (with learning rate 0.0002, and exponential decay rate 0.5) for the Generator (G) and SDG (with learning rate 0.01) for the Authenticator (A). The noise vector used by the Generator G is based on a standard Gaussian noise. In all experiments, we use training batches of size 100.

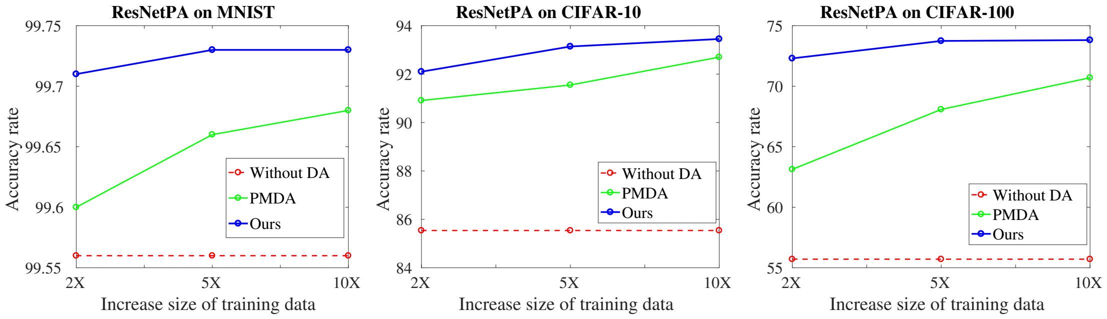
Comparison results using ResNet18 and ResNetpa networks are shown in Figures 3.3 and 3.4. First, in all cases it is clear that DA provides a significant improvement in the classification accuracy – in general, larger augmented training set sizes lead to more accurate classification. More importantly, the results reveal that our Bayesian DA algorithm outperforms PMDA by a large margin in all datasets. Given the similarity between the model used by our proposed Bayesian

[3] The code was adapted from: `https://github.com/lukedeo/keras-acgan`



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

Figure 3.3: Performance comparison using ResNet18 [36] classifier.

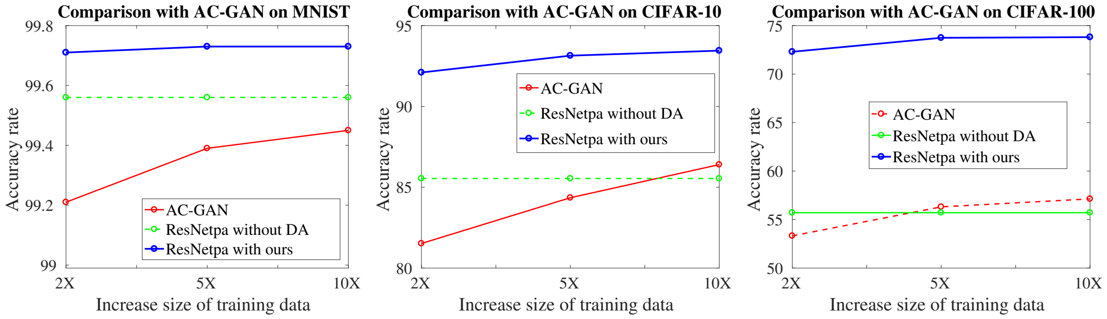


(a) MNIST (b) CIFAR-10 (c) CIFAR-100

Figure 3.4: Performance comparison using ResNetpa [37] classifier.

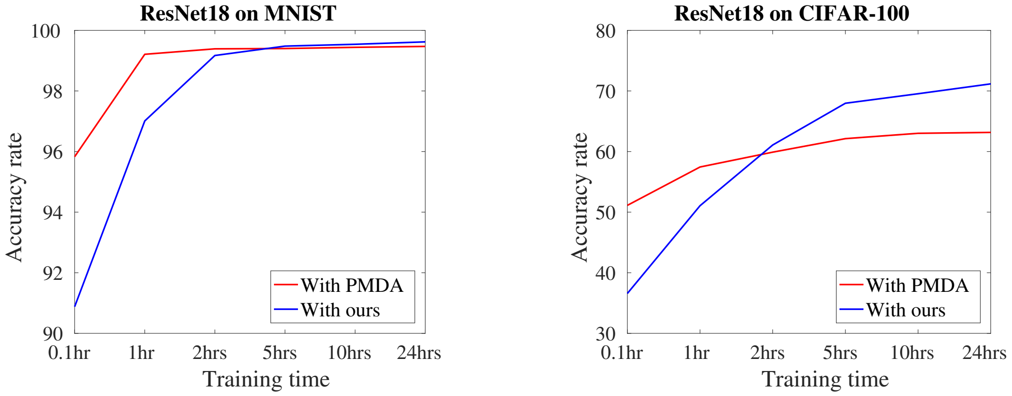
DA algorithm (using ResNetpa [37]) and AC-GAN, it is relevant to present a comparison between these two models, which is shown in Fig. 3.5 – notice that our approach is far superior to AC-GAN. Finally, it is also important to show the evolution of the test classification accuracy as a function of training time – this is reported in Fig. 3.6. As expected, it is clear that PMDA produces better classification results at the first training stages, but after a certain amount of training, our Bayesian DA algorithm produces better results. In particular, using the ResNet18 [36] classifier, on CIFAR-100, our method is better than PMDA after two hours of training; while for MNIST, our method is better after five hours of training.

It is worth emphasizing that the main goal of the proposed Bayesian DA is to improve the training process of the classifier C. Nevertheless, it is also of interest to investigate the quality of the images produced by the generator G. In Fig. 3.7,



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

Figure 3.5: Performance comparison with AC-GAN using ResNetpa [37]



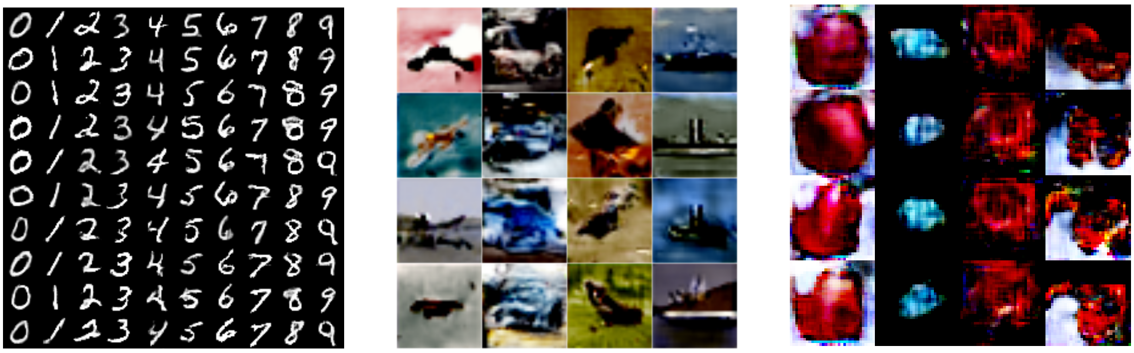
(a) MNIST (b) CIFAR-100

Figure 3.6: Classification accuracy (as a function of the training time) using PMDA and our proposed data augmentation on ResNet18 [36]

we display several examples of the synthetic images produced by G after the training process has converged. In general, the images look reasonably realistic, particularly the handwritten digits, where the synthesized images would be hard to generate by the application of Gaussian or uniform noise on pre-determined geometric and appearance transformations.

3.6 Conclusions

In this chapter we have presented a novel Bayesian DA that improves the training process of deep learning classification models. Unlike currently dominant methods



(a) MNIST

(b) CIFAR-10

(c) CIFAR-100

Figure 3.7: Synthesized images generated using our model trained on MNIST (a), CIFAR-10 (b) and CIFAR-100 (c). Each column is conditioned on a class label: a) classes are $0, ..., 9$; b) classes are airplane, automobile, bird and ship; and c) classes are apple, aquarium fish, rose and lobster.

that apply random transformations to the observed training samples, our method is theoretically sound; the missing data are sampled from the distribution learned from the annotated training set. However, we do not train the generator distribution independently from the training of the classification model. Instead, both models are jointly optimised based on our proposed Bayesian DA formulation that connects the classical latent variable method in statistical learning with modern deep generative models. The advantages of our data augmentation approach are validated using several image classification tasks with clear improvements over standard DA methods and also over the recently proposed AC-GAN model [74].

Acknowledgments

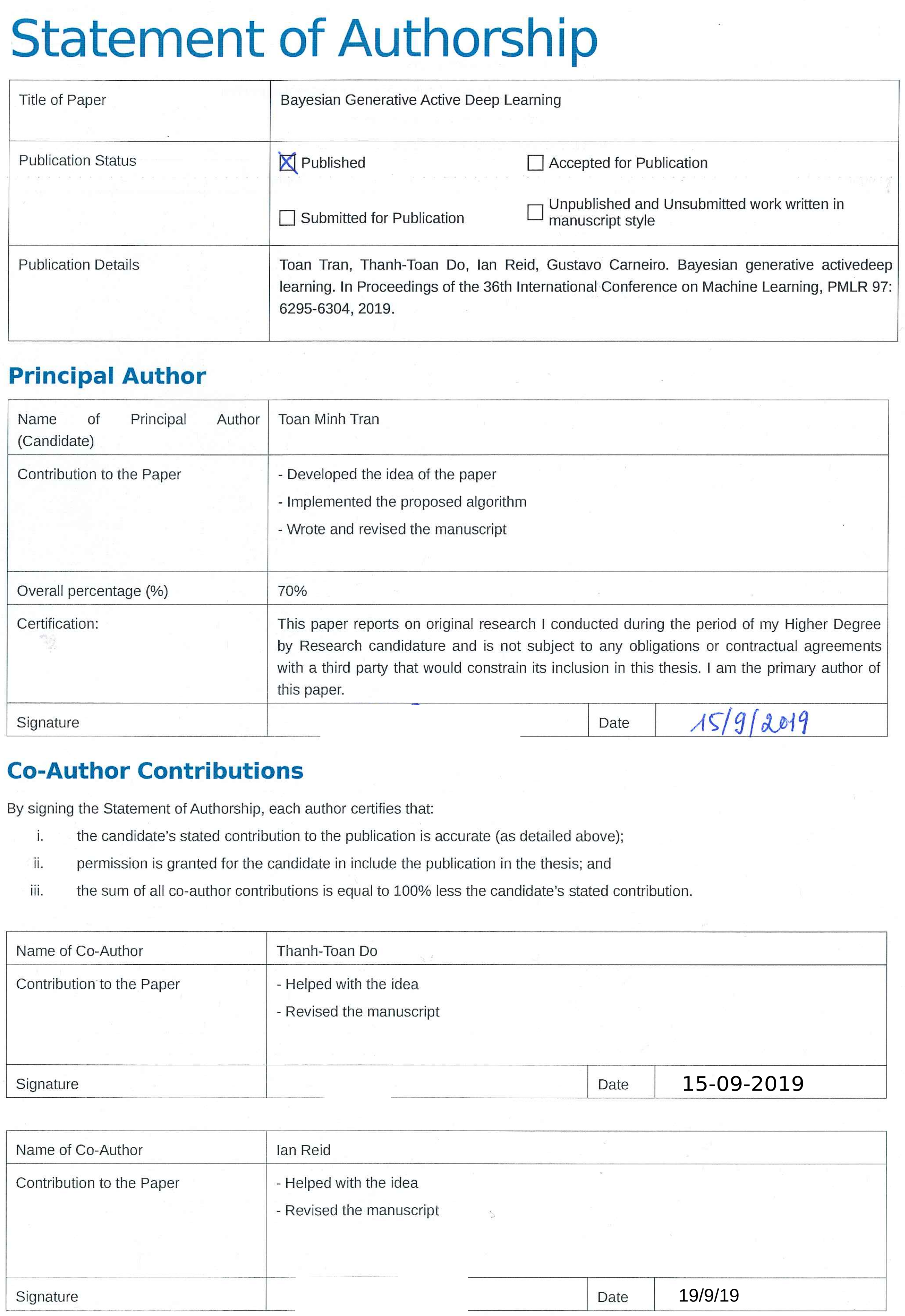
TT gratefully acknowledges the support by Vietnam International Education Development (VIED). TP, GC and IR gratefully acknowledge the support of the Australian Research Council through the Centre of Excellence for Robotic Vision (project number CE140100016) and Laureate Fellowship FL130100102 to IR.

CHAPTER 4

Bayesian Generative Active Deep Learning

The work contained in this chapter has been published as the following paper:

T. Tran, T.-T. Do, I. Reid, and G. Carneiro. Bayesian generative active deep learning. In *International Conference on Machine Learning*, pages 6295–6304, 2019. [98]



Statement of Authorship

Title of Paper	Bayesian Generative Active Deep Learning	
Publication Status	☒ Published □ Submitted for Publication	□ Accepted for Publication □ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Toan Tran, Thanh-Toan Do, Ian Reid, Gustavo Carneiro. Bayesian generative activedeep learning. In Proceedings of the 36th International Conference on Machine Learning, PMLR 97: 6295-6304, 2019.	

Principal Author

Name of Principal Author (Candidate)	Toan Minh Tran		
Contribution to the Paper	- Developed the idea of the paper - Implemented the proposed algorithm - Wrote and revised the manuscript		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	15/9/2019

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Thanh-Toan Do		
Contribution to the Paper	- Helped with the idea - Revised the manuscript		
Signature		Date	15-09-2019

Name of Co-Author	Ian Reid		
Contribution to the Paper	- Helped with the idea - Revised the manuscript		
Signature		Date	19/9/19

Name of Co-Author	Gustavo Carneiro		
Contribution to the Paper	- Developed the idea of the paper - Suggested some ideas to implement the proposed algorithm - Supervised the development of the work - Wrote and revised the manuscript		
Signature		Date	15-09-2019

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 labelling, limiting the types of problems that can be tackled. Therefore, the design of effective training methods that require small labelled 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 unlabelled 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.

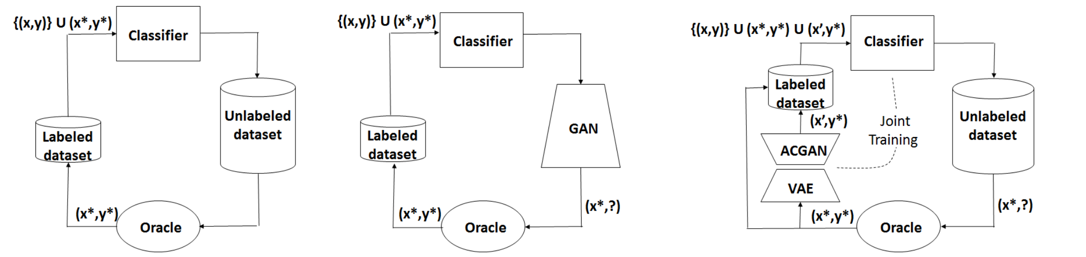
4.1 Introduction

Deep learning is undoubtedly the dominant machine learning methodology [23, 43, 54, 77]. Part of the reason behind this success lies in its training process that can be performed with immense and carefully labelled data sets, where the larger the data set, the more effective the training process [92]. However, the labelling of such large data sets demands significant human effort, and the large-scale training process requires considerable computational resources [92]. These training issues have prevented researchers and practitioners from solving a wider range of classification problems, where large labelled data sets are hard to acquire or vast computational resources are unavailable [65]. Addressing these issues is one of the most important problems to be solved in machine learning [3,26,48,53,85,99,108].

One of the most successful approaches to mitigate the issue described above relies on the use of a small labelled data set and a large unlabelled data set, where small subsets from the unlabelled set are automatically selected using an acquisition function that assesses how informative those subsets are for the training process. These selected unlabelled subsets are then labelled by an oracle (i.e., a human annotator), integrated into the labelled 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 [85], and its aim is to reduce the need for large labelled 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 unlabelled data set does not exist, then a possible idea is to use the samples from the labelled 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) [53] or that have been estimated from a generative adversarial training [99]. This training process is known as data augmentation, which targets the reduction of the need for large labelled 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 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 labelled data set with generated samples that are informative for the training process – see Fig. 4.1. Our paper is motivated by the following works: query by synthesis active learning [108], Bayesian data augmentation [99], auxiliary-classifier generative adversarial networks (ACGAN) [74] and variational autoencoder (VAE) [49]. We assume the existence of a small labelled and a large unlabelled data set, where we use the



a) Pool-based Active Learning b) Generative Adversarial Active Learning c) Bayesian Generative Active Deep Learning

Figure 4.1: Comparison between (pool-based) active learning [85] (a), generative adversarial active learning [108] (b), and our proposed Bayesian generative active deep learning (c). The labelled data set is represented by $\{(\mathbf{x}, \mathbf{y})\}$, the unlabelled point to be labelled 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 $\mathbf{x}'$.

concept of Bayesian active learning by disagreement (BALD) [26, 42] to select informative samples from the unlabelled data set. These samples are then labelled 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 labelled data set to be used in the next training iteration.

Compared to a recently proposed generative adversarial active learning [108], which relies on an optimisation problem to generate new samples (this optimisation balances sample informativeness with image generation quality), our approach has the advantage of using acquisition functions that have proved to be more effective [26] than the simple information loss in [108]. Different from our approach that trains the generative and classification models jointly, the approach in [108] 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 unlabelled 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 unlabelled data set to guarantee the informativeness of the selected samples [34]. 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 unlabelled 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 [61], CIFAR-$\{10, 100\}$ [52] and SVHN [72].

4.2 Related Work

4.2.1 Bayesian Active Learning

In a (pool-based) active learning framework, the learner is initially modelled with a small labelled training set, and it will iteratively search for the "most informative" samples from a large unlabelled data set to be labelled by an oracle – these newly labelled samples are then used to re-model the learner. The information value of an unlabelled sample is estimated by an acquisition function, which is maximised in order to select the most informative samples. For example, the most informative samples can be selected by maximising the "expected informativeness" [67], or minimising the "expected error" of the learner [11] – such acquisition functions are hard to optimise 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. [42] 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. [26] 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 [25] to approximate this and other commonly used acquisition functions. This approach [26] 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.

4.2.2 Data Augmentation

In active learning, it is assumed that a model can be trained to achieve an acceptable level of accuracy with a small data set. That assumption is challenging in the estimation of a deep learning model since it often requires large labelled data sets to avoid over-fitting. One reasonable way to increase the labelled training set is with data augmentation that artificially generates new synthetic training samples [53]. Gal et al. [26] also emphasised the importance of data augmentation for the development of deep active learning. Data augmentation can be performed with "label-preserving" transformations [53, 89, 104] – this is known as "poor's man" data augmentation (PMDA) [94, 99]. Alternatively, Bayesian data augmentation (BDA) trains a deep generative model (using the training set), which is then used to produce new artificial training samples [99]. Compared to PMDA, BDA has been shown to have a better theoretical foundation and to be more beneficial in practice [99]. 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. 4.2.

4.2.3 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 unlabelled pool, Zhu & Bento [108] 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 optimisation during generation is solved

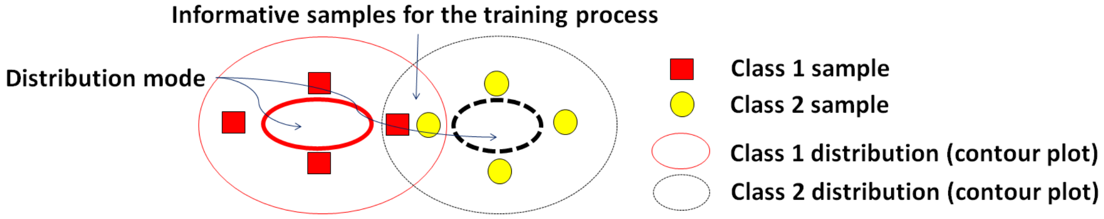

Informative samples for the training process
Distribution mode
Class 1 sample
Class 2 sample
Class 1 distribution (contour plot)
Class 2 distribution (contour plot)

Figure 4.2: 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" [85]. These generated instances are informative for the training process since the learning model is uncertain about them [42].

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 optimise (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 [26]. Also, the GAN model in [108] 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 unlabelled 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 neighbourhood. 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.

4.2.4 Variational Autoencoder Generative Adversarial Networks

Generative Adversarial Network (GAN) [30] 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 [70] and ACGAN [74]. In order to tackle the low diversity problem (known as "mode collapse"), Larsen et al. [58] introduced a variational autoencoder generative adversarial network (VAE-GAN) that combines a variational autoencoder (VAE) [49] and a GAN in which these networks are connected by a generator/decoder [107]. We utilise both ACGAN and VAE-GAN in our proposed Bayesian Generative Active Deep Learning framework, but with the aim of improving the classification performance.

4.3 "Information-Preserving" Data Augmentation for Active Learning

4.3.1 Bayesian Active Learning by Disagreement (BALD)

Let us denote the initial labelled 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 labelled with $\mathbf{y}_i \in \mathcal{C} = \{1, 2, \ldots, C\}$, where C is the number of 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 [42], the most informative sample $\mathbf{x}^*$ is selected from the (unlabelled) pool data $\mathcal{D}_{\text{pool}}$ by [42]:

$$\begin{aligned}
\mathbf{x}^* &= \arg\max_{\mathbf{x} \in \mathcal{D}_{\text{pool}}} a(\mathbf{x}, \mathcal{M}) \\
&= \arg\max_{\mathbf{x} \in \mathcal{D}_{\text{pool}}} H[\mathbf{y}|\mathbf{x}, \mathcal{D}] - \mathbb{E}_{\theta \sim p(\theta|\mathcal{D})}[H[\mathbf{y}|\mathbf{x}, \theta]], \qquad (4.1)
\end{aligned}$$

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 [87] 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 labelled with $\mathbf{y}^*$ (by an oracle), and the labelled 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 (4.1), Gal et al. [26] introduced the Monte Carlo (MC) dropout method. This objective function can be approximated by its sample mean [26]:

$$a(\mathbf{x}, \mathcal{M}) \approx -\sum_{c=1}^{C} \left(\frac{1}{T} \sum_{t=1}^{T} \hat{p}_c^t \right) \log \left(\frac{1}{T} \sum_{t=1}^{T} \hat{p}_c^t \right) + \frac{1}{T} \sum_{c=1}^{C} \sum_{t=1}^{T} \hat{p}_c^t \log \hat{p}_c^t, \tag{4.2}$$

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.

4.3.2 Generative Model and Bayesian Data Augmentation

In the iterative Bayesian data augmentation (BDA) framework [99], 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 [99], the authors also showed a weak convergence proof that is related to the improvement of the posterior distribution $p(\theta|\mathcal{D})$.

4.3.3 Bayesian Generative Active Deep Learning

The main technical contribution of this paper consists of combining BALD and BDA for generating new labelled samples that are informative for the training process (see Fig. 4.3).

We modify BDA [99] 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 (4.1) using the estimation (4.2) is pushed to go through a variational autoencoder (VAE) [49], which contains an encoder $e(.)$ and a decoder $g(.)$, in order to generate the sample $\mathbf{x}'$, as follows:

$$\mathbf{x}' = g(e(\mathbf{x}^*)). \tag{4.3}$$

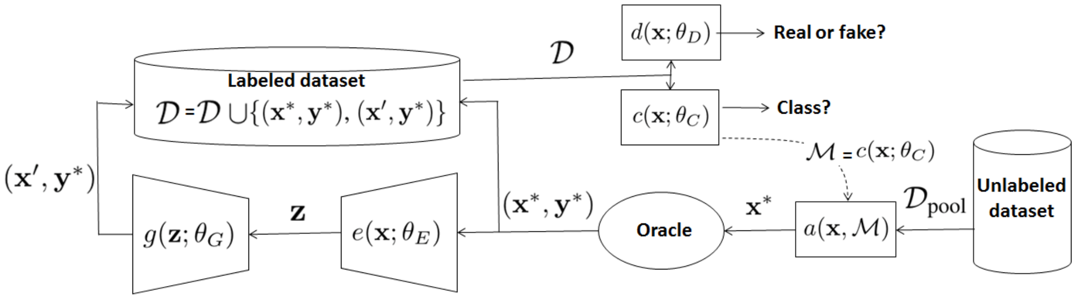

$d(\mathbf{x};\theta_D)$
Real or fake?
$\mathcal{D}$
Labeled dataset
$\mathcal{D}=\mathcal{D}\cup\{(\mathbf{x}^*,\mathbf{y}^*),(\mathbf{x}',\mathbf{y}^*)\}$
$c(\mathbf{x};\theta_C)$
Class?
$\mathcal{M}=c(\mathbf{x};\theta_C)$
$(\mathbf{x}',\mathbf{y}^*)$
$\mathbf{z}$
$g(\mathbf{z};\theta_G)$
$e(\mathbf{x};\theta_E)$
$(\mathbf{x}^*,\mathbf{y}^*)$
Oracle
$\mathbf{x}^*$
$a(\mathbf{x},\mathcal{M})$
$\mathcal{D}_{\text{pool}}$
Unlabeled dataset

Figure 4.3: Network architecture of our proposed model.

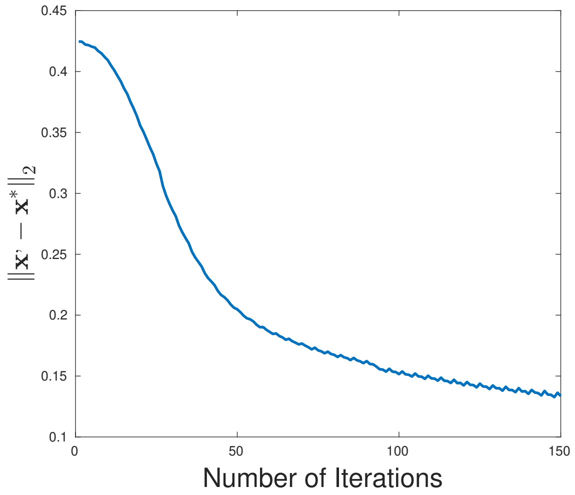


Figure 4.4: Reduction of $\|\mathbf{x}' - \mathbf{x}^*\|$ as the training of the VAE model progresses (on CIFAR-100 using ResNet-18).

The training process of a VAE is performed by minimising the "reconstruction loss" $\ell(\mathbf{x}^*, g(e(\mathbf{x}^*))$ [49], where if the number of training iterations is sufficiently large, we have:

$$\|\mathbf{x}' - \mathbf{x}^*\| < \varepsilon, \tag{4.4}$$

with ε representing an arbitrarily small positive constant – see Fig. 4.4 for an evidence for that claim.

The label of $\mathbf{x}'$ is assumed to be $\mathbf{y}^*$ (i.e., the oracle's label for $\mathbf{x}^*$) and the current labelled 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 2. 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 (5.5) is also informative.

Proof. Given the assumptions of Proposition 2, and due to the fact that $\mathbf{x}^*$ is a local maximum of function $a(\mathbf{x}, \mathcal{M})$ (4.1), then $\mathbf{x}^*$ is a critical point of $a(\mathbf{x}, \mathcal{M})$, i.e.,

$$\nabla_x a(\mathbf{x}^*, \mathcal{M}) = \mathbf{0}. \tag{4.5}$$

Condition (5.6), which is empirically verified by Fig. 4.4, indicates that $\mathbf{x}'$ belongs to a sufficiently small neighbourhood 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 (5.7), we obtain

$$\begin{aligned} a(\mathbf{x}', \mathcal{M}) &\approx a(\mathbf{x}^*, \mathcal{M}) + \nabla_x a(\mathbf{x}^*, \mathcal{M})^T (\mathbf{x}' - \mathbf{x}^*) \\ &\approx a(\mathbf{x}^*, \mathcal{M}), \end{aligned} \tag{4.6}$$

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

4.4 Implementation

Our network, depicted in Fig. 4.3, 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 [36,37,61], 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 [74] and VAE-GAN [58], 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 [58,107] was formed by adding the reconstruction error in VAE to the GAN loss in order to penalise both *unrealisticness* and *mode collapse* in GAN training. Following that, the VAE-ACGAN loss of our proposed model is defined by

$$\mathcal{L} = \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{ACGAN}}, \tag{4.7}$$

with the VAE loss [49,58] represented by a combination of the reconstruction loss $\mathcal{L}_{\text{rec}}$ and the regularisation prior $\mathcal{L}_{\text{prior}}$, i.e.,

$$\begin{aligned}\mathcal{L}_{\text{VAE}} &= \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{prior}} \\ &= \mathcal{L}_{\text{rec}}(\mathbf{x}, g(e(\mathbf{x};\theta_E);\theta_G) + D_{\text{KL}}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})),\end{aligned} \tag{4.8}$$

where $\mathbf{z} = e(\mathbf{x};\theta_E)$, $p(\mathbf{z})$ 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 loss [74] in (4.7) is computed by

$$\begin{aligned}\mathcal{L}_{\text{ACGAN}} &= \log(d(\mathbf{x};\theta_D)) + \log(1 - d(g(\mathbf{z};\theta_G);\theta_D)) \\ &+ \log(1 - d(g(\mathbf{u};\theta_G);\theta_D)) + \log(\text{softmax}(c(\mathbf{x};\theta_C))) \\ &+ \log(\text{softmax}(c(g(\mathbf{z};\theta_G);\theta_C))) \\ &+ \log(\text{softmax}(c(g(\mathbf{u};\theta_G);\theta_C))),\end{aligned} \tag{4.9}$$

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

4.5 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) [99] (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 [61], CIFAR-10, CIFAR-100 [53], and SVHN [72]. MNIST [61] contains handwritten digits, (with 60,000 training and 10,000 testing samples, and 10 classes), CIFAR-10 [53] is composed of 32 × 32 colour images (with 50,000 training and 10,000 testing samples, and 10 classes), CIFAR-100 [53] is similar to CIFAR-10,

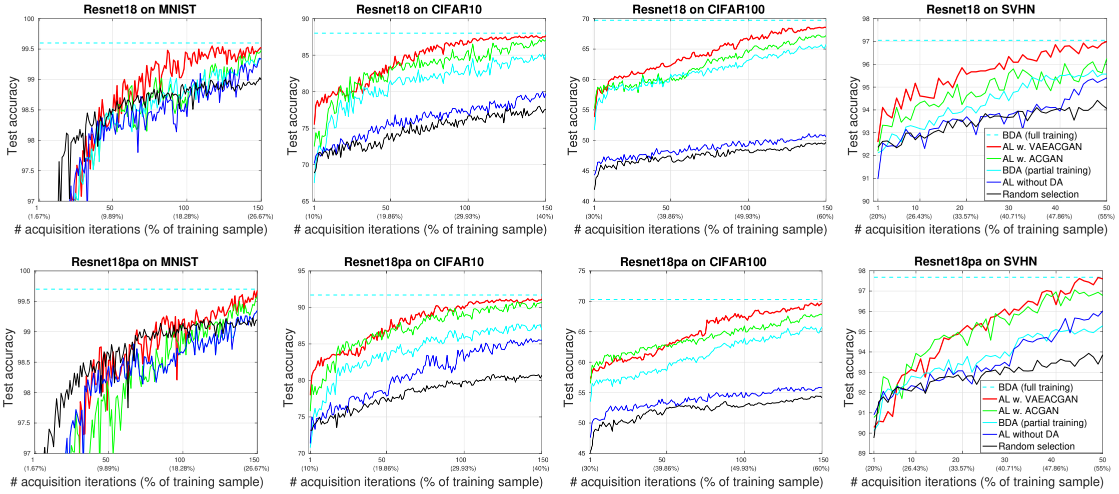
[1] code available at `https://github.com/toantm/BGADL`

Algorithm 1 Bayesian Generative Active Learning

```
Initialise 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 (4.1) and (4.2), 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 $\gamma = 0.75$ [58] 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
```

but with 100 classes, and SVHN [72] 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 [36] and ResNet18pa [37], 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 unlabelled data set $\mathcal{D}_{\text{pool}}$ (this fixed number of randomly selected samples almost certainly contains the most informative sample [34]). 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 optimiser with (lr=0.0002, $\beta_1 = 0.5$, $\beta_2 = 0.999$); the mini-batch size is 100 for all cases.



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

Figure 4.5: Training and classification performance of the proposed Bayesian generative active learning (*AL w. VAEACGAN*) compared to active learning using BDA [99] (*AL w. ACGAN*), BDA modelled with partial training sets (*BDA (partial training)*), BALD [26, 42] 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 modelled with the full training set (*BDA (full training)*) and 10× 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 modelling. First row shows these results using ResNet18 [36], and second row shows ResNet18pa [37] on MNIST [61] (column 1), CIFAR-10 (column 2) CIFAR-100 [53] (column 3), and SVHN [72] (column 4).

Fig. 4.5 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 modelling. The methods compared are: 1) BDA [99] modelled with the full training set (*BDA (full training)*) and 10× 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 modelled with partial training sets (*BDA (partial training)*); 5) BALD [26, 42] 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. 4.5 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. 4.5, *BDA (partial training)* relies on 2× 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.

Table 4.1: Mean ± standard deviation of the classification accuracy on MNIST, CIFAR-10, and CIFAR-100 after 150 iterations over 3 runs

	MNIST					
	AL w. VAEACGAN	AL w. ACGAN	AL w. PMDA	AL without DA	BDA (partial training)	Random selection
Resnet18	**99.53 ± 0.05**	99.45 ± 0.02	99.37 ± 0.15	99.33 ± 0.10	99.33 ± 0.04	99.00 ± 0.13
Resnet18pa	**99.68 ± 0.08**	99.57 ± 0.07	99.49 ± 0.09	99.35 ± 0.11	99.35 ± 0.07	99.20 ± 0.12
	CIFAR-10					
Resnet18	**87.63 ± 0.11**	86.80 ± 0.45	82.17 ± 0.35	79.72 ± 0.19	85.08 ± 0.31	77.29 ± 0.23
Resnet18pa	**91.13 ± 0.10**	90.70 ± 0.24	87.70 ± 0.39	85.51 ± 0.21	86.90 ± 0.27	80.69 ± 0.19
	CIFAR-100					
Resnet18	**68.05 ± 0.17**	66.50 ± 0.63	55.24 ± 0.57	50.57 ± 0.20	65.76 ± 0.40	49.67 ± 0.52
Resnet18pa	**69.69 ± 0.13**	67.79 ± 0.76	59.67 ± 0.60	55.82 ± 0.31	65.79 ± 0.51	54.77 ± 0.29

To show a more informative comparison of our proposed approach (*AL w. VAEACGAN*) with other methods presented in Fig. 4.5, especially with *AL w. ACGAN* and *BDA (partial 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 ± stdev) in Tab. 4.1 (after 150 iterations).

We also compare the average information value of samples measured by the acquisition function (4.2) of the samples generated by *AL w. ACGAN* and *AL w. VAEACGAN* in Fig. 4.6 using Resnet18 on CIFAR-100.

Figure 4.7 displays images generated by our generative model for each data set.

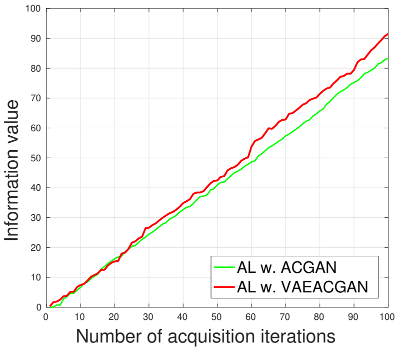
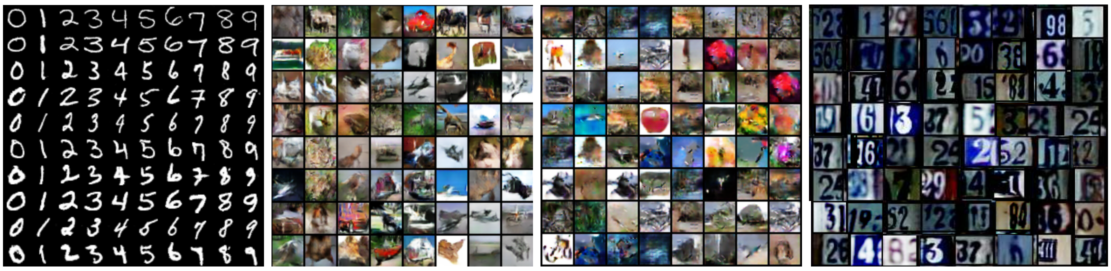


Figure 4.6: Average information value of samples measured by the acquisition function (4.2) of the samples generated by *AL w. ACGAN* and *AL w. VAEACGAN* using Resnet18 on CIFAR-100.



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

Figure 4.7: Images generated by our proposed *AL w. VAEACGAN* approach for each data set.

4.6 Discussion and Conclusions

Results in Fig. 4.5 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. 4.6 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 4.1 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. 4.5 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× data augmentation over the entire training set – this is evidence that the generation of informative training samples can use less human and computer resources for labelling 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. 4.5 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. 4.5 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. 4.7, 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 [108]. Although relevant to our approach, [108] 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 [21, 27, 84]. In the future, we plan to investigate how to generate samples directly using complex acquisition functions, such as the one in (4.2), instead of conditioning the sample generation on highly informative samples selected from the unlabelled data set. We also plan to work on the efficiency of our proposed method because its empirical computational cost is slightly higher than BDA [99] and BALD [26, 42].

Acknowledgements

We gratefully acknowledge the support by Vietnam International Education Development (VIED), Australian Research Council through grants DP180103232, CE140100016 and FL130100102.

CHAPTER 5

Bayesian Generative Active Deep Learning Applied to Imbalanced Learning

The work contained in this chapter has been submitted as the following paper:

Toan Tran, Ian Reid, Gustavo Carneiro. Bayesian Generative Active Deep Learning Applied to Imbalanced Learning. Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2019.

Statement of Authorship

Title of Paper	Bayesian Generative Active Deep Learning Applied to Imbalanced Learning	
Publication Status	□ Published ☒ Submitted for Publication	□ Accepted for Publication □ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Toan Tran, Ian Reid, Gustavo Carneiro. Bayesian Generative Active Deep Learning Applied to Imbalanced Learning. Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 2019.	

Principal Author

Name of Principal Author (Candidate)	Toan Minh Tran		
Contribution to the Paper	- Developed the idea of the paper - Implemented the proposed algorithm - Wrote and revised the manuscript		
Overall percentage (%)	70%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	25-09-2019

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Ian Reid		
Contribution to the Paper	- Helped with the idea - Revised the manuscript		
Signature		Date	26/9/19

Name of Co-Author	Gustavo Carneiro		
Contribution to the Paper	- Developed the idea of the paper - Suggested some ideas to implement the proposed algorithm - Supervised the development of the work - Wrote and revised the manuscript		
Signature		Date	26/09/2019

Abstract

This chapter extends our recently proposed Bayesian generative active deep learning framework proposed in chapter 4 that aims to improve data efficiency in training a deep learning model. In particular, the goal of our original method is to generate new synthetic training data points that are informative for the training process. The algorithm consists of an efficient combination of deep Bayesian active learning and Bayesian data augmentation, in which the active selection scheme in active learning is used to guide the generator of the Bayesian data augmentation to generate novel informative training samples. We extend the proposed Bayesian generative active deep learning method to work with imbalanced learning problems by combining it with a sample re-weighting scheme. Experiments on MNIST, CIFAR-10, and SVHN show a significant improvement of the Bayesian generative active deep learning approach compared to other related approaches. Furthermore, experiments with imbalanced data sets indicate that the extension of the proposed method can perform well on imbalanced training data.

5.1 Introduction

Deep learning has been shown to be a dominant machine learning approach that can improve the state-of-the-art in speech recognition, computer vision, and many other domains such as drug discovery and genomics [29,54,60,77]. One of the key training issues in deep learning is that it often requires not only a large amount of carefully labelled training samples [92], but also a well-balanced class distribution in the training data [1,46,68]. Seeking solutions to handle these training issues is essential for researchers and practitioners since that can help reduce the number of training samples and computational resources for training deep models and enable the use of imbalanced training sets. Relevant to this paper, we can identify three learning approaches to improve data efficiency in training deep models: active learning [42, 85], data augmentation [53, 99], and hybrid methods that combine these two approaches [51,98,108]. To address the class imbalance data problem, reasonable solutions involve a re-balancing of the skewed data set by re-sampling [8,46] or sample re-weighting [78] methods.

Active learning [32,42,85] was motivated by the fact that the large amount of unlabelled data is cheap to acquire while obtaining labelled data is much more expensive. The goal of active learning is to achieve a certain model performance using as less labelled training data as possible. Active learning generally consists of an iterative learning scheme, where the learner is initially modelled with a small annotated data set, and it then actively requests the most informative samples to be labelled (by an oracle) and trained upon. The active selection of the most informative samples is performed by maximising an acquisition function that evaluates the usefulness of unlabelled data samples for the training process. Such classical active learning has been shown to reduce the amount of necessary training samples (i.e., the sample complexity) compared to other traditional passive learning methods [26, 32, 98]. However, this active learning approach is challenging to be directly applied in the estimation of a deep learning model since at the beginning of the training process, the active learner is likely to over-fit the small initial training sets [98].

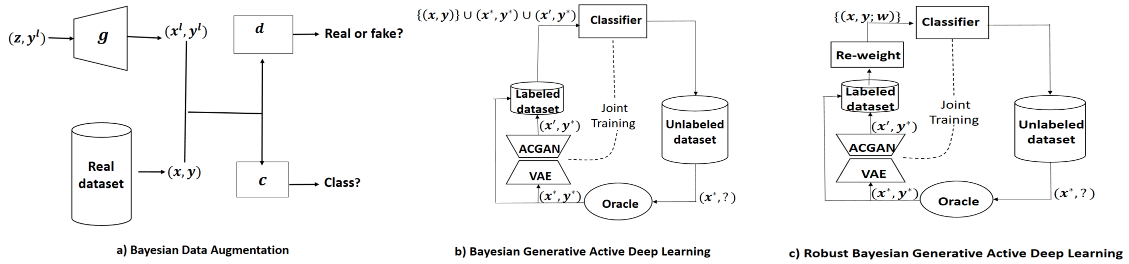
If the unlabelled data set is unavailable or difficult to access, then one reasonable alternative is to artificially enlarge the existing data sets to avoid over-fitting the training set. That approach is known as data augmentation (DA), which can lead to a robust training of a deep learning model [53,99]. One of the key benefits of data augmentation is that it can avoid manually labelling training samples, which is often time-consuming, subjective and prone to mistakes [99]. The data augmentation can be performed by using several small-scale linear transformations such as random rotation, translation or colour perturbation in order to preserve the ground truth label of the real sample [53] – this is referred to as the "poor man's" data augmentation (PMDA) [95, 99]. Although useful in practice [53, 89, 104], the strong assumption about the label-preserving small-scale transformations does not provide any guarantee that PMDA generates useful training samples [99]. That is, it can generate unrealistic samples and it can fail to produce realistic samples.

We first proposed in [99] a novel theoretically sound data augmentation, namely Bayesian data augmentation (BDA) that targets the generation of novel synthetic samples learned from the likelihood of the data given the model parameter. BDA [99], which was inspired by the data augmentation using latent variable method [95], is trained by a variant of the expectation maximisation (EM) algorithm [18], called generalised Monte-Carlo expectation maximisation

(GMCEM). The key benefit of data augmentation is that it can generate immense amount of artificial data points to target a more robust training of a deep learning model. However, BDA tends to waste not only training time but also computational resources since the generation of new samples is done without regarding the informativeness (usefulness) of the generated samples [98].

In an attempt to address the issues mentioned above regarding active learning and BDA, we proposed a Bayesian generative active deep learning framework that aimed to generate informative samples for the training process [98]. That algorithm consists of a theoretically sound combination of the Bayesian active learning by disagreement (BALD) [26, 42] and BDA [99], in which the active selection step is employed to guide the data augmentation scheme to generate informative training data points. One potential drawback of the generative active deep learning in [98] is that it was not designed to handle imbalanced training problems. That imbalanced data issue occurs when some classes, in the majority group, contain significantly more training samples than other classes, in the minority group – such problems appear in many real-world applications, such as cancer classification and fraud detection [46], protein fold classification and weld flaw classification [103]. Imbalanced learning can make model classification less effective due to poor predictions on minority classes, but such classification tends to be important in typical imbalanced learning problems (e.g., it is generally more important to avoid false negative than false positive classification in cancer diagnosis) [103]. Anand et al. [1] pointed out that the backpropagation algorithm-based training process of a neural network can get stuck since the gradient can be dominated by the majority classes' gradient components. To handle imbalanced learning, one of the most popular ways is to re-balance the original imbalanced data set by using several random re-sampling schemes such as under-sampling the majority class or over-sampling the minority class [46]. Recently, Ren et al. [78] proposed a novel robust meta-learning method that aims to learn to re-weight the training samples without changing the size of the original training set.

In this paper, we first present the two learning methods proposed in our previous papers, that are Bayesian data augmentation [99], and Bayesian generative active deep learning [98]. In particular, we provide a more insightful literature review, formulations, mathematical justifications for these two methods. To address the class imbalance issue that may occur in the newly updated labelled training data at each iteration of the Bayesian generative active deep learning [98], we then extend that learning method to introduce a novel learning approach that is robust against class imbalance data by combining the Bayesian generative active learning framework with the sample re-weighting approach [78]. In particular, we use the sample re-weighting method [78] to re-balance the newly updated labelled training data set at each active learning iteration in the Bayesian generative active deep learning scheme [98]–as depicted in Fig. 5.1.



a) Bayesian Data Augmentation

b) Bayesian Generative Active Deep Learning

c) Robust Bayesian Generative Active Deep Learning

Figure 5.1: Comparison between our proposed methods in previous papers: Bayesian data augmentation (BDA) [99], Bayesian generative active deep learning [98] and our novel proposed Robust Bayesian generative active deep learning against imbalanced data.

Experimental results on MNIST, CIFAR-10 and SVHN show that the proposed Bayesian generative active deep learning improves over other related approaches. Furthermore, classification performance on three imbalanced data subsets sampled from MNIST, CIFAR-10 and SVHN show a considerable improvement of our proposed method compared to other baselines.

5.2 Related Work

In this section, we explore current literature, and analyse several research gaps in some relevant methods that will be addressed in our proposed approaches. We first provide a brief description of the (pool-based) active learning framework, and one of its extensions for deep learning, called Bayesian active learning by disagreement (BALD). We then discuss the concept of the dominant "poor man's"data augmentation (PMDA), and the motivation of our proposed Bayesian data augmentation method (BDA) [99]. We next introduce several generative active learning schemes, including our proposed Bayesian generative active deep learning [98]. We also discuss generative adversarial network (GAN) and its variants that are used in some of the implementations of our proposed algorithms. Finally, we investigate imbalanced learning methods to motivate our novel Bayesian generative active deep learning algorithm that is robust to class imbalance data.

5.2.1 Active Learning

In a general (pool-based) active learning framework [32, 85], the model is initially trained with a small labelled training set. Then, it automatically selects a subset of the most informative unlabelled samples from the pool data to be annotated (by an oracle) – these newly labelled informative samples are then added to the original training data set for the next modelling iteration. The active selection of the most informative instances can be performed, for example, by maximising an acquisition function that can be evaluated by the "expected informativeness" [67], or the (negative) "expected error" of the learner [11]. Optimising these acquisition functions is challenging in deep learning due to the computational complexity of the inverse of the Hessian matrix of the expected error with respect to the high-dimensional model parameter [98].

Targetting an efficient acquisition function estimation in deep active learning, Houlsby et al. [42] investigated the Bayesian active learning by disagreement (BALD) scheme, in which the active learner aims to seek for unlabelled samples from the pool data such that the current model parameters (under the posterior distribution) vigorously disagrees about their labels [42]. That BALD algorithm is also known as the "information theoretic active learning" method, in which the acquisition function is estimated by the "mutual information" of the label of the sample with respect to the model parameters. Gal et al. [26] then introduced the Monte-Carlo dropout method to estimate the BALD acquisition function and several other types of acquisition functions that can be employed in active deep learning. Recently, Kirsch et al. [50] strengthened the BALD method by extending that acquisition function, called BatchBALD, to improve the data diversity in the mini-batch of informative samples. A common problem among the methods above is that the estimation of a deep learning model in active learning may lead to over-fitting since that model is assumed to rely on a small informative training set. One reasonable way is to enlarge the given labelled data set by generating novel synthetic training data points [53]. That approach is known as data augmentation, which has been widely employed in several computer vision tasks [53], and is explained in the section below.

5.2.2 Data Augmentation

The dominant data augmentation used in the field is known as the "poor man's" data augmentation (PMDA) [95,99], where the generation of new artificial training samples is executed only once, prior to the training process, and is performed by using several sufficiently small scale linear transformations [53,89,104] to preserve the labels of the real training samples. One of the challenging problems in using PMDA lies in how to choose the group of transformations to optimise the model performance. Cubuk et al. [16] and Lim et al. [64] extended PMDA by combining it with reinforcement learning to seek for an optimal subset of the given sets of the linear transformations. Although PMDA has been shown to work well in practice, it has not been properly validated and may not generate realistic samples [99].

In an attempt to investigate a novel theoretically sound data augmentation, we proposed in [99] a Bayesian data augmentation (BDA) that aims to train a generative model to produce new training data points. In particular, we formulate the BDA framework based on a variant of the expectation maximisation (EM) algorithm, called generalised Monte-Carlo expectation maximisation (GMCEM). We theoretically proved the weak convergence, which is related to an improvement of the posterior distribution at each parameter estimation step, of that GMCEM framework. We also introduced in [99] an implementation of BDA based on ACGAN [74] – that is a variant of generative adversarial network (GAN) [30]. In particular, our implementation adapts ACGAN [74] by using the generator and separating the classifier and the discriminator. That separation makes our BDA scheme more flexible since it allows us to use different discriminative models, targeting at an improvement of the classification performance [99]. The key difference of BDA compared to PMDA is that in BDA, the generator and the model are jointly optimised, and the generation of the novel synthetic samples can therefore be adaptively learned as the training progresses [99].

5.2.3 Generative Active Learning

If the unlabelled data pool is not available, then one reasonable way is to generate novel informative synthetic data points to accelerate the training performance. This method is known as generative active learning, which aims to train a generative model to produce informative samples for the training process. Zhu and Bento [108] proposed the generative adversarial active learning (GAAL) method to generate novel informative artificial samples with a pre-trained GAN model [30]. Kong et al. [51] then extended the GAAL method with a new uncertainty reward to guide a conditional GAN to generate new informative training data points. The key advantage of these methods is that they can generate informative data points for the training process without the requirement of the unlabelled pool data set, given that the GAN model is pre-trained and the optimisation problem can be solved efficiently. Nevertheless, this approach is challenging to be applied in deep learning because they usually require overly simple acquisition functions, such as the (negative) distance from the sample to the hyper-plane [85, 108] – these functions do not in general represent well the complexity of active deep learning problems [26, 98].

One of the drawbacks of both GAAL approaches above is that they only focus on the binary classification problem. We proposed in [98] a Bayesian generative active deep learning–an efficient combination of Bayesian data augmentation (BDA) [99] and Bayesian active learning by disagreement (BALD) [26, 42], that also targets the generation of informative training data points. In contrast to the two methods above [51, 108], our proposed Bayesian generative active deep learning [98] can handle multi-class problems using a deep classifier. Moreover, at each iteration of that Bayesian generative active deep learning, the learner and the generator are jointly optimised in the sample optimisation problem–this training principle allows them to "co-evolve" during the training process [98]. In particular, at each training iteration of that Bayesian generative active deep learning [98], the most informative training sample selected by maximising the acquisition function over the pool data set is then processed through a VAE-ACGAN [98] model to generate novel synthetic data point. The current labelled training set is then augmented by both of these selected and generated samples for the next training iteration. We theoretically show in [98] that the new generated synthetic sample is also informative for the training process due to a small difference between the value of the acquisition function at that sample and the optimal acquisition value (i.e., the value at the selected most informative one).

5.2.4 GANs and VAEs

Generative adversarial network (GAN) [30] is an influential deep generative model that aims to learn how to generate new artificial training samples from observed data [29]. More specifically, GAN estimates a generative model through an "adversarial process" performed by simultaneously training two deep learning models: a generator that learns to produce samples belonging to a good approximation of the (unknown) ground truth data distribution, and a discriminator that maximises the probability that a sample came from the true data distribution rather than from the generator. The performance of a GAN model is commonly evaluated by the inception score (IS) [82] and the Frechet inception score (FID) [38] that relate both to the quality and the diversity of the generated samples. In order to improve the synthetic image quality, Odena et al. [74] introduced ACGAN, which is then modified to demonstrate the Bayesian data augmentation algorithm in [99].

One of the most challenging problems that affects the diversity of the generated images in the training of a GAN model is "mode collapse"–this issue occurs when a real sample has significantly small probability to be generated. In an attempt to address that "mode collapse" problem, Larsen et al. [58] proposed a variational autoencoder generative adversarial network (VAE-GAN) that is a combination of VAE [49] and GAN, in which these generative models are linked by the decoder/generator [107]. In our previous work [98], we adapted that combined network to introduce the VAE-ACGAN model in a demonstration of the Bayesian generative active deep learning [98] by conditioning the generator on the selected informative sample and its label to generate a new synthetic training data point that is also informative for the training process.

5.2.5 Imbalanced Learning

Effective learning with imbalanced data is crucial. Imbalanced learning can happen in many scenarios, such as in medical diagnosis, where the majority group is usually healthy, but the performance on the minority class is more critical [46]. One of the most popular ways to address class imbalance is to re-balance the original imbalanced training data set by under-sampling the majority class or over-sampling the minority class [34, 46]. For example, in imbalanced active learning [34], Ertekin [22] proposed the virtual instance re-sampling technique using active learning (VIRTUAL) to re-balance data set by generating informative minority samples–that VIRTUAL method focuses only on the binary classification problem. One of the key drawbacks of those random re-sampling approaches is that under-sampling can reduce the target information value of the model, while over-sampling tends to increase the training time and computational resources, which can even lead to over-fitting [8,46]. To re-balance a skewed data set without changing its size, Ren et al. [78] proposed a novel robust meta-learning method that aims to learn to re-weight the training samples by minimising the weighted loss function associated to the performance on a balanced validation set. We then use this sample re-weighting method [78] to handle the class imbalance issue that may appear in the newly updated training data set in the Bayesian generative active deep learning framework [98].

5.3 Methodology

In this section we provide comprehensive formulations and mathematical justifications for the following methods: Bayesian active learning by disagreement (BALD) [42]; Bayesian data augmentation [99]; Bayesian generative active deep learning [98]; and meta sample re-weighting approach for imbalanced learning [78]. We then extend the Bayesian generative active deep learning [98] to introduce a novel method that is robust to class imbalance data by combining that with the sample re-weighting approach [78].

5.3.1 Bayesian Active Learning by Disagreement (BALD)

Let us denote the observed data by $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathcal{X}$ is the data sample and its corresponding label $\mathbf{y}_i \in \mathcal{C} = \{1, 2, \ldots, C\}$, where C is the number of classes, and the model $\mathcal{M}$ is parameterised by θ. This model $\mathcal{M}$ is represented by a Bayesian neural network, which means that the modelling process is based on the estimation of the posterior distribution of the parameters θ given $\mathcal{D}$, namely $p(\theta|\mathcal{D})$.

We formulate the estimation of $p(\theta|\mathcal{D})$ based on the Bayesian active learning by disagreement (BALD) scheme [26, 42, 98]. In each iteration of BALD, the most informative sample $\mathbf{x}^*$ is selected from the (unlabelled) data pool $\mathcal{D}_{\text{pool}}$ by maximising the acquisition function $a(\mathbf{x}, \mathcal{M})$ that can be approximated with Monte Carlo (MC) dropout method [26], i.e.,

$$a(\mathbf{x}, \mathcal{M}) \approx -\sum_{c=1}^{C} \left(\frac{1}{D} \sum_{d=1}^{D} \hat{p}_c^d \right) \log \left(\frac{1}{D} \sum_{d=1}^{D} \hat{p}_c^d \right) + \frac{1}{D} \sum_{c=1}^{C} \sum_{d=1}^{D} \hat{p}_c^d \log \hat{p}_c^d, \tag{5.1}$$

where D is the number of dropout iterations, $\hat{\mathbf{p}}^d = [\hat{p}_1^d, \ldots, \hat{p}_C^d] = \text{softmax}(f(\mathbf{x}; \theta^d))$, with f denoting the network function parameterised by θ^d sampled from an approximation of the posterior $p(\theta|\mathcal{D})$ at the d-th iteration. That sample $\mathbf{x}^*$ is then annotated by an oracle, producing $(\mathbf{x}^*, \mathbf{y}^*)$, which is added to the annotated data set for the next iteration of BALD (see Fig. 5.2).

5.3.2 Generative Models and Bayesian Data Augmentation

The estimation of a deep model using BALD in Sec. 5.3.1 may lead to over-fitting since that model is assumed to rely on a small informative training set. One reasonable way to avoid that over-fitting issue is to use data augmentation to enlarge the given labelled data set by generating novel synthetic training data points [53]. In the dominant data augmentation approach, the generation of the new artificial training samples can be performed by using several sufficiently small scale linear transformations [53, 89, 104] to preserve the labels of the real samples. Although this approach is shown to work well in practice, that data augmentation method, which we refer to as "poor man's" data augmentation (PMDA) [95], has not been properly validated. For instance, it is still unclear if

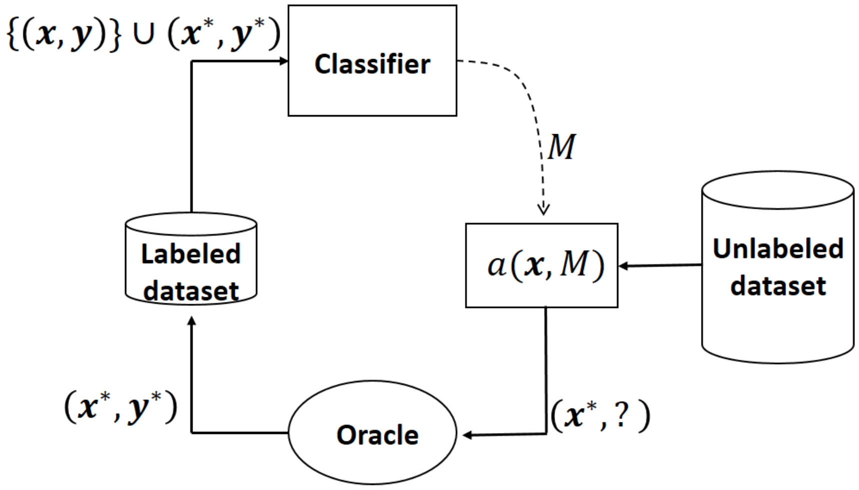


Figure 5.2: Bayesian Active Learning by Disagreement (BALD) [42,98].

small scale linear transformations actually preserve the labels of the sample, and if such range of transformations provides good coverage of the image variations for a particular visual classification problem. We introduce a novel theoretically sound data augmentation, proposed in [99], called Bayesian data augmentation (BDA) to train a generative model to re-generate new training data points.

The Bayesian data augmentation (BDA) scheme [99] is formulated based on a generalised Monte-Carlo expectation maximisation (GMCEM) – a variant of the expectation maximisation (EM) algorithm [18,95,99]. Each iteration of GMCEM consists of two steps: 1) synthetic data generation and expectation approximation (E-step); and 2) model training (M-step). This synthesised data is then inserted in a set of latent variables defined by $\mathcal{D}^l = \{\mathbf{x}^l, \mathbf{y}^l\}_{l=1}^{|\mathcal{D}^l|}$, and the augmented data set is represented by $\mathcal{D}^a = \mathcal{D} \cup \mathcal{D}^l$. The E-step of the BDA algorithm consists of computing a Monte-Carlo approximation $\hat{Q}(\theta, \theta^t)$ of $Q(\theta, \theta^t)$ that is the expectation of the log of augmented posterior $\log p(\theta|\mathcal{D}^a)$ [99]:

$$Q(\theta, \theta^t) = \mathbb{E}_{(\mathbf{x}^l, \mathbf{y}^l) \sim p(\mathcal{D}^l|\theta^t, \mathcal{D})} \log p(\theta|\mathcal{D}^a), \tag{5.2}$$

where θ^t is the estimation at the t-th iteration of the model parameter θ. The M-step of BDA comprises the estimation of θ^{t+1} with:

$$\theta^{t+1} = \arg\max_{\theta} Q(\theta, \theta^t). \tag{5.3}$$

The algorithm iterates until $\|\theta^{t+1} - \theta^t\|$ is sufficiently small, and the optimal θ^* is selected from the final iteration. The EM steps of BDA in (5.2) and (5.3) guarantee that:

$$\hat{Q}(\theta^{t+1}, \theta^t) > \hat{Q}(\theta^t, \theta^t). \tag{5.4}$$

The weak convergence related to a true posterior improvement (i.e., $p(\theta^{t+1}|\mathcal{D}) > p(\theta^t|\mathcal{D})$)) of the GMCEM framework is guaranteed by the following lemma that we proposed in a previous paper [99]:

Lemma 3. *Assuming that θ^{t+1} is obtained by an iteration of the EM algorithm, i.e., $\hat{Q}(\theta^{t+1}, \theta^t) \geq \hat{Q}(\theta^t, \theta^t)$, which is guaranteed from* (5.4)*, then the weak convergence (i.e. $p(\theta^{t+1}|\mathbf{y}) > p(\theta^t|\mathbf{y})$) will be fulfilled.*

Proof. Given $\hat{Q}(\theta^{t+1}, \theta^t) > \hat{Q}(\theta^t, \theta^t)$, then by taking the expectation on both sides, that is $\mathbb{E}_{p(\mathbf{z}|\mathbf{y},\theta^t)}[\hat{Q}(\theta^{t+1}, \theta^t)] > \mathbb{E}_{p(\mathbf{z}|\mathbf{y},\theta^t)}[\hat{Q}(\theta^t, \theta^t)]$, we obtain $Q(\theta^{t+1}, \theta^t) > Q(\theta^t, \theta^t)$, which is the condition for $p(\theta^{t+1}|\mathbf{y}) > p(\theta^t|\mathbf{y})$ proven from [95]. □

We propose an implementation of the BDA [99] based on an ACGAN model [74] (see Fig. 5.1-(a)). In particular, our implementation extends ACGAN [74] by using the generator and separating the classifier from the discriminator. In other words, it consists of a generator $g(\cdot)$ (that generates new synthetic training samples), a discriminator $d(\cdot)$ (that identifies real and fake samples) and a classification model $c(\cdot)$ (that classifies samples in one of the classes in $\mathcal{C}$). At each training iteration, a novel synthetic labelled data point $(\mathbf{x}^l, \mathbf{y}^l) \in \mathcal{D}^l$ is generated by $\mathbf{x}^l = g(\mathbf{z}, \mathbf{y}^l)$, where the noise $\mathbf{z}$ is often chosen as a multivariate Gaussian, and the label $\mathbf{y}^l \in \mathcal{C}$. The loss function related to the training process of that three-subnetwork model consists of two parts: the discriminative loss defined by $J_c + J_{dg}$, and the generative loss defined by $J_c - J_{dg}$, where J_c is the optimisation function of the classifier c, and J_{dg} is the one used to train the discriminator and the generator networks [99]. We also theoretically show in [99] that that joint loss function is linked to the objective function in (5.4), and therefore, the (weak) convergence of the training procedure using stochastic gradient descent can be guaranteed [99].

5.3.3 Bayesian Generative Active Deep Learning

The generation of samples from BDA, explained in Sec. 5.3.2, synthesises $(\mathbf{x}^l, \mathbf{y}^l)$ given a noise vector $\mathbf{z}$ sampled from a multivariate Gaussian. Such approach guarantees that the synthesised samples belong to $\mathcal{D}^l$ that approximates the true data distribution $\mathcal{D}$ from the training of our ACGAN-extended model. However, it does not guarantee that the generated sample is informative for the next iteration process. In fact, as training progresses, it is expected that the synthesised samples become less informative given that most of them will be based on samples $\mathbf{z}$ that are close to the mean of the multivariate Gaussian. Therefore, we propose a new approach that guarantees the generation of informative samples, resulting in a training approach that requires much fewer annotated training samples.

Our proposed approach is the generative active deep learning method [98] that consists of two steps (see Fig. 5.1-(b)): 1) active sample selection, and 2) informative synthetic sample generation. In the first step, the most informative training data point $\mathbf{x}^*$ is selected by maximising the approximated acquisition function in (5.1). That selected sample is then processed by a VAE-ACGAN [98] model, which is modified from the Bayesian data augmentation [99] to generate new training data sample $\mathbf{x}'$,

$$\mathbf{x}' = g(e(\mathbf{x}^*)), \tag{5.5}$$

where $e(\cdot)$ is the encoder and $g(\cdot)$ is the decoder of the VAE and also the generator of the ACGAN model. The newly generated sample $\mathbf{x}'$ is shown to belong to a sufficiently small neighbourhood of the most informative sample $\mathbf{x}^*$ due to the deduction of the reconstruction loss in the VAE training [98], i.e., with an arbitrarily small positive constant δ, we have

$$\|\mathbf{x}' - \mathbf{x}^*\| < \delta, \tag{5.6}$$

and $\mathbf{x}'$ therefore can inherit the label $\mathbf{y}^*$ of $\mathbf{x}^*$ (the label is provided by an oracle). Both of these samples, $(\mathbf{x}^*, \mathbf{y}^*)$ and $(\mathbf{x}', \mathbf{y}^*)$ are then added to the original labelled data set, i.e., $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{x}^*, \mathbf{y}^*), (\mathbf{x}', \mathbf{y}^*)$, which is used for the next training iteration. More importantly, the informativeness of the generated data point $\mathbf{x}'$ is theoretically guaranteed by the following proposition [98]:

Proposition 4. 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 (5.5) is also informative.

Proof. Given the assumptions of Proposition 4, and due to the fact that $\mathbf{x}^*$ is a local maximum of function $a(\mathbf{x}, \mathcal{M})$ (5.1), then $\mathbf{x}^*$ is a critical point of $a(\mathbf{x}, \mathcal{M})$, i.e.,

$$\nabla_x a(\mathbf{x}^*, \mathcal{M}) = \mathbf{0}. \tag{5.7}$$

Condition (5.6) indicates that $\mathbf{x}'$ belongs to a sufficiently small neighbourhood 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 (5.7), we obtain

$$\begin{aligned} a(\mathbf{x}', \mathcal{M}) &\approx a(\mathbf{x}^*, \mathcal{M}) + \nabla_x a(\mathbf{x}^*, \mathcal{M})^T (\mathbf{x}' - \mathbf{x}^*) \\ &\approx a(\mathbf{x}^*, \mathcal{M}), \end{aligned} \tag{5.8}$$

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

5.3.4 Imbalanced Learning

The generative active deep learning from Sec. 5.3.3 does not handle imbalanced distribution of samples per class since the augmentation of the selected informative and the generated training samples is done without regarding how balanced the class distribution of the newly updated training set is. This section extends our previously proposed method [98] by extending it to work in imbalanced problems. The idea is to use the learn to re-weight method [78] to re-balance the updated training data set at each iteration of the generative active learning framework [98].

One reasonable solution to address that class imbalance issue is to re-balance the original imbalanced training data set by under-sampling the majority class or over-sampling the minority class [34,46]. However, under-sampling can reduce the target information value of the model, while over-sampling tends to increase the training time and computational resources, and can even lead to over-fitting [8,46].

Alternatively, Ren et al. [78] proposed a meta-learning scheme that learns to re-weight the data samples without changing the training data size. The key idea of that re-weighting process is to minimise a weighted loss [78]:

$$\theta^*(\mathbf{w}) = \arg\min_{\theta} \sum_{i=1}^{N} w_i \ell(f(\mathbf{x}_i;\theta), \mathbf{y}_i), \tag{5.9}$$

where $\ell(\cdot,\cdot)$ is the loss function, and the optimal weight vector $\mathbf{w} = (w_1, \ldots, w_N)$ is estimated by minimising the corresponding performance on a balanced validation set $\mathcal{D}^v = \{\mathbf{x}_j^v, \mathbf{y}_j^v\}_{j=1}^M$ [78], i.e.,

$$\mathbf{w}^* = \arg\min_{\mathbf{w},\mathbf{w}\geq 0} \frac{1}{M} \sum_{j=1}^{M} \ell(f(\mathbf{x}_j^v;\theta^*(\mathbf{w})), \mathbf{y}_j^v). \tag{5.10}$$

The authors in [78] also introduced an online approximation frame work to improve the computational complexity of solving (5.10) by adapting online $\mathbf{w}$ at each optimisation iteration. In particular, the weight vector $\mathbf{w}$ is first restricted to be in the set $\{\mathbf{w} : \ \|\mathbf{w}\|_1 = 1\} \cup \{0\}$, and its i-th element at t-th training iteration, namely $w_{i,t}$ can be defined by [78]:

$$w_{i,t} = \frac{\tilde{w}_{i,t}}{(\sum_j \tilde{w}_{j,t}) + \delta(\sum_j \tilde{w}_{j,t})}, \tag{5.11}$$

where $\delta(a) = \begin{cases} 1, & \text{if } a = 0 \\ 0, & \text{otherwise} \end{cases}$, and

$$\begin{aligned} \tilde{w}_{i,t} &= \max(u_{i,t}, 0), \\ u_{i,t} &= -\eta \frac{\partial}{\partial \varepsilon_{i,t}} \frac{1}{M} \sum_{j=1}^{M} \ell(f(\mathbf{x}_j^v;\theta_{t+1}(\varepsilon)), \mathbf{y}_j^v)|_{\varepsilon_{i,t}=0}, \\ \theta_{t+1}(\varepsilon) &= \theta_t - \alpha \nabla \sum_{i=1}^{N} \varepsilon_i \ell(f(\mathbf{x}_i;\theta), \mathbf{y}_i)|_{\theta=\theta_t}, \end{aligned}$$

where the last equation refers to the $(t+1)^{th}$ update of model parameters θ.

5.3.5 Generative Active Deep Learning Robust to Imbalanced Learning Algorithm

This section introduces the main contribution of this paper that extends our previously proposed method [98] to work for imbalanced learning problems (see

Fig. 5.1-(c)). Assuming that the initial training data set $\mathcal{D}$ is imbalanced, the idea is to use the sample re-weighting meta-learning approach [78] to re-balance the skewed data set without changing the data set size. The learning algorithm relies on the weight w_i of the sample $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}$ (defined in Eq. 5.9) to form the weighted loss to be minimised. At each iteration of the generative active deep learning [98], the initial training set $\mathcal{D}$ is augmented by two novel training data points including the selected most informative sample $(\mathbf{x}^*, \mathbf{y}^*)$ and the informative generated sample $(\mathbf{x}', \mathbf{y}^*)$, i.e., $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{x}^*, \mathbf{y}^*), (\mathbf{x}', \mathbf{y}^*)\}$. These samples in the updated training set are then re-weighted with the sample re-weighting procedure [78] for the next training iteration.

We modify the Bayesian generative active deep learning [98] to propose a novel training process, described in Algorithm 2, that is robust to class imbalanced data sets. In that algorithm the balanced validation set $\mathcal{D}^v$ is used for the re-weighting procedure.

5.4 Experiments and Results

In this section, we first show and discuss the experiments for balanced data sets [98] to show the better classification performance of the Bayesian generative active deep learning approach [98] compared to other baseline methods. We then evaluate our proposed generative active deep learning robust to imbalanced learning in Sec. 5.3.5. For all experiments, we rely on the top-1 classification accuracy in a comparison with several baselines.

5.4.1 Experiments on Balanced Data Sets [98]

We present the classification performance as a function of the number of acquisition iterations and the respective percentage of training samples used in a particular training iteration. We present results of the following methods: 1) 10× data augmentation BDA [99] trained on the full training set (*BDA (full training)*); 2) the Bayesian generative active learning (*AL w. VAEACGAN*); 3) active learning using BDA (*AL w. ACGAN*); 4) 2× data augmentation BDA modelled with partial training sets (*BDA (partial training)*); 5) BALD [26,42] without data augmentation (*AL without DA*); and 6) random selection of training samples using the percentage

Algorithm 2 Robust Bayesian Generative Active Learning

```
Employ the sample re-weighting method [78] to obtain the weight vector $\mathbf{w}$ on the initial training set $\mathcal{D}$, using the validation set $\mathcal{D}^v$.
Initialise 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 (5.1), where $\mathcal{M}$ is represented by the classifier $c(\mathbf{x};\theta_C)$;
    Request the oracle to label the selected sample, which forms $(\mathbf{x}^*, \mathbf{y}^*)$
    Re-weight the updated training set using the weight vector $\mathbf{w}$, defined above
    $\mathbf{z} \leftarrow e(\mathbf{x}^*;\theta_E)$
    $\mathcal{L}_{\text{prior}} \leftarrow D_{\text{KL}}(q(\mathbf{z}|\mathbf{x}^*)\|p(\mathbf{z}))$
    $\mathbf{x}' = g(e(\mathbf{x}^*);\theta_G)$
    $\mathcal{L}_{\text{rec}} \leftarrow \mathcal{L}_{\text{rec}}(\mathbf{x}^*, \mathbf{x}')$
    Assign the learned weight of $\mathbf{x}^*$ for the weight of $\mathbf{x}'$
    Sample $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathcal{L}_{\text{ACGAN}} \leftarrow \log(d(\mathbf{x}^*)) + \log(1 - d(\mathbf{x}')) + \log(1 - d(g(\mathbf{u}))) + \log(\text{softmax}(c(\mathbf{x}^*;\mathbf{w}))) + \log(\text{softmax}(c(\mathbf{x}';\mathbf{w}))) + \log(\text{softmax}(c(g(\mathbf{u}))))$
    $\theta_E \leftarrow \theta_E - \nabla_{\theta_E}(\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{prior}})$
    $\theta_G \leftarrow \theta_G - \nabla_{\theta_G}(\gamma\mathcal{L}_{\text{rec}} - \mathcal{L}_{\text{ACGAN}})$ (parameter $\gamma = 0.75$ [58] 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
```

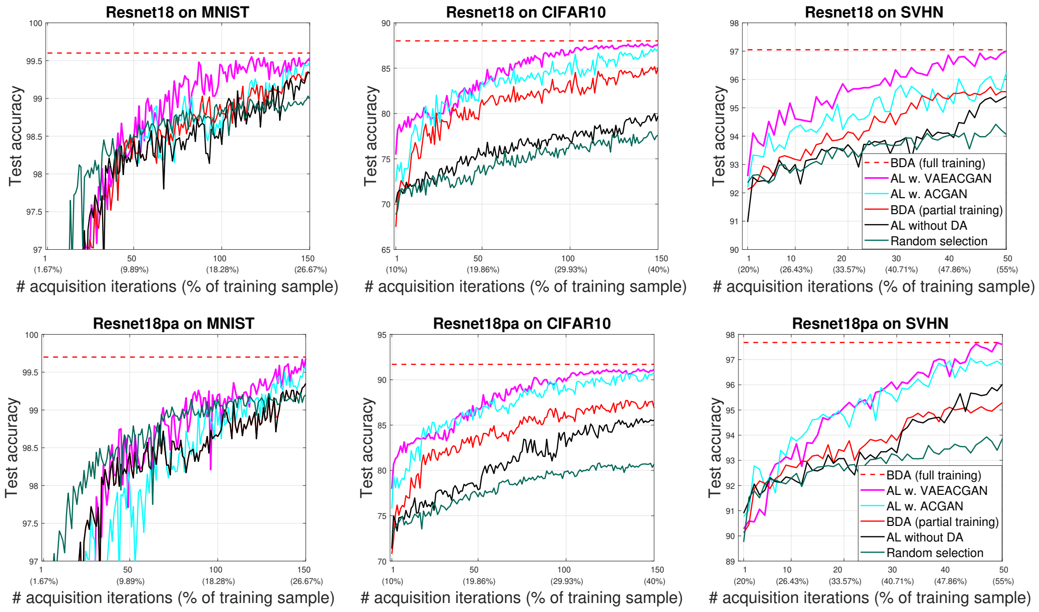
of samples from the original training set (*Random selection*). The comparison is based on experiments using two classifiers, ResNet18 [36] and ResNet18pa [37], on the following benchmark data sets: MNIST [61], CIFAR-10 [53], and SVHN [72]. MNIST [61] contains 28×28 black and white handwritten digits, (with 60000 training and 10000 testing samples, and 10 classes), CIFAR-10 [53] is composed of 32×32 colour images (with 50000 training and 10000 testing samples, and 10 classes), and SVHN [72] contains 32×32 colour images containing street view house numbers (with 73257 training samples and 26032 testing samples, and 10 classes).

In [98], the sample acquisition setup for each data set is chosen as follows: 1) the number of samples in the initial training set is 1000 for MNIST, 5000 for CIFAR-10, and 10000 for SVHN (this initial data set size was empirically determined based on the fact that with smaller sizes, 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 2000 randomly selected samples of the unlabelled data set $\mathcal{D}_{\text{pool}}$. The hyper-parameters of the training process are chosen as follows: 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. 5.3 shows the comparison with respect to the classification accuracy as a function of both number of acquisition iterations and corresponding proportion of training samples.

5.4.2 Experiments on Imbalanced Data Sets

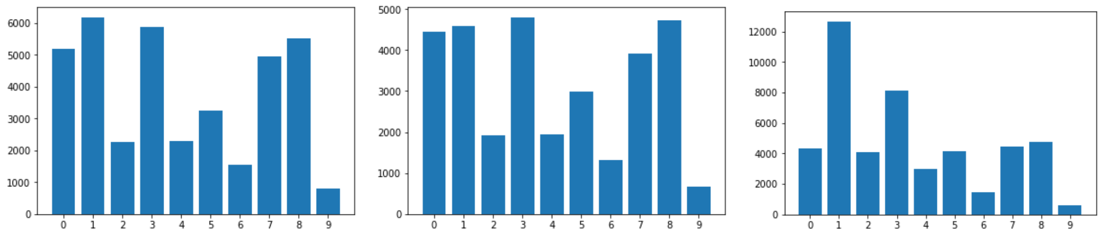
In this section, our proposed Bayesian generative active deep learning robust to imbalanced data sets (*AL w. VAEACGAN using RW*) is compared with the following baseline approaches: 2) Bayesian generative active deep learning without sample re-weighting (*AL w. VAEACGAN without RW*); 3) AL w. ACGAN using sample re-weighting (*AL w. ACGAN using RW*); 4) AL w. ACGAN without sample re-weighting (*AL w. ACGAN without RW*); 5) active learning using sample re-weighting without data augmentation (*AL using RW*); 6) active learning without data augmentation nor sample re-weighting (*AL*); 7) sample re-weighting



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

Figure 5.3: Training and classification performance of the proposed Bayesian generative active learning (*AL w. VAEACGAN*) compared to active learning using BDA [99] (*AL w. ACGAN*), BDA modelled with partial training sets (*BDA (partial training)*), BALD [26,42] 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 modelled with the full training set (*BDA (full training)*) and 10× 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 modelling. First row shows these results using ResNet18 [36], and second row shows ResNet18pa [37] on MNIST [61] (column 1), CIFAR-10 (column 2), and SVHN [72] (column 3).

(*RW*) modelled on three training subsets associated with the first acquisition, the 50^{th}-acquisition iteration, and the final acquisition iteration. That comparison is based on experiments performed on imbalanced subsets randomly sampled from MNIST [61], CIFAR-10 [53], and SVHN [72]. In particular, the imbalanced MNIST contains 37829 training samples with the corresponding number of samples per class: [5214, 6201, 2329, 5880, 2283, 3259, 1522, 4939, 5485, 717]; the imbalanced CIFAR-10 consists of 31323 training samples with the corresponding number of samples per class: [4375, 4570, 1922, 4800, 1978, 3000, 1298, 4001, 4697, 682]; the imbalanced SVHN contains 47482 training samples with the corresponding number of samples per class: [4389, 12628, 4023, 8157, 2937, 4122, 1439, 4377, 4733, 677]–The class distribution of each imbalanced data set is depicted in Fig. 5.4.



(a) Imbalanced MNIST (b) Imbalanced CIFAR-10 (c) Imbalanced SVHN

Figure 5.4: Distribution of classes in imbalanced data sets used in the experiments in Sec. 5.4.2 for (a) MNIST, (b) CIFAR-10, and (c) SVHN.

To demonstrate the flexibility of our proposed method, we test it with three deep models: Lenet [62], Resnet18 [36] and Resnet18pa [37]. The setup for this experiment is slightly different from the one used for the balanced data set experiment regarding the number of samples in the initial data sets, and the number of acquisitions. We needed to change this setup because of differences observed in the structure of imbalanced data sets. In particular, the sizes of the initial training sets are: a) for the imbalanced MNIST, 100 for Lenet, 1000 for Resnet18 and Resnet18pa; b) for imbalanced CIFAR-10, 5000 for all three models; and c) for the imbalanced SVHN, 5000 for all three models. For all experiments, the number of acquisition iterations is 100, and at each acquisition function, 100 samples are selected from a subset that contains 2000 samples of the unlabelled pool data set $\mathcal{D}_{\text{pool}}$. We use the same hyper-parameters defined in [98], which is also mentioned in Sec. 5.4.1. The balanced validation set in the sample re-weighting algorithm contains 1000 samples (100 for each class) randomly chosen from the original balanced data sets in Sec. 5.4.1 – this validation set does not contain samples belonging to the corresponding imbalanced data set.

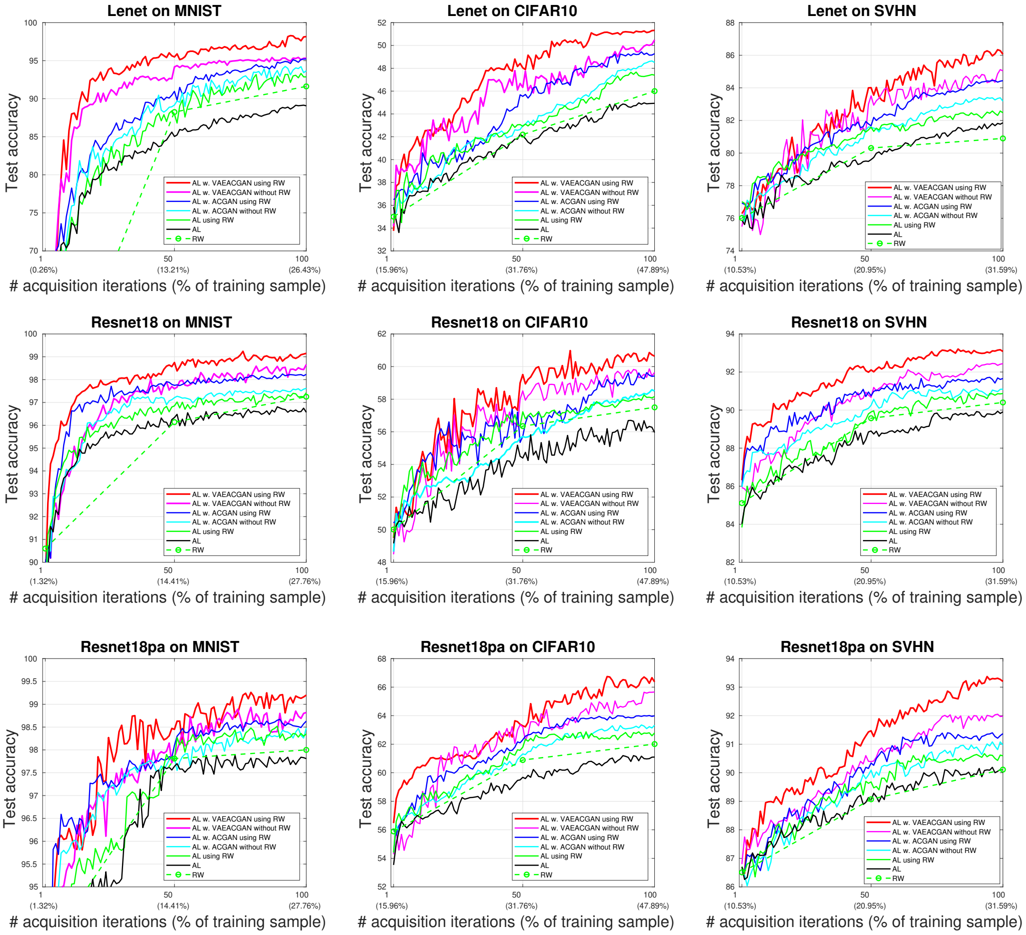
In our proposed Bayesian generative active deep learning robust to imbalanced data, detailed in Alg. 2, the newly updated labelled training set is re-weighted with the sample re-weighting scheme [78] at every acquisition. This makes the training algorithm computationally expensive. To handle that issue, in our actual experiments that are performed on the labelled data sets, we use the sample re-weighting scheme only once on the whole imbalanced data set. The learned weight is then stored and assigned to the corresponding selected informative sample in order to form the weighted loss function at each training iteration.

All experimental results are presented in Fig. 5.5.

5.5 Discussion and Conclusion

It is clear from Fig. 5.3 that, in all experiments for balanced data sets, our proposed Bayesian generative active deep learning (*AL w. VAEACGAN*) [98] provides better classification accuracy than active learning with *BDA* (*AL w. ACGAN*), in which the informative samples are also used for training, but the generated samples may not be informative [98]. However, such training samples generated with *AL w. ACGAN* seem to be crucial given its superior results, compared with active learning without data augmentation (*AL without DA*). Moreover, results in Fig. 5.3 also indicate the efficiency of the generation of informative samples given the similar classification accuracy produced by *AL w. VAEACGAN* and 10× *BDA (full training)*, but *AL w. VAEACGAN* uses a fraction of the training set of 10× *BDA* (e.g., 26.67%, 40%, and 55% for MNIST, CIFAR-10, and SVHN, respectively). Furthermore, it can be seen from Fig. 5.3 that given training sets with the same number of training samples, *AL w. VAEACGAN* outperforms *BDA* (partial training) by a larger margin across different data sets and classifiers [98].

The classification results in Fig. 5.5 consistently show that our proposed Bayesian generative active deep learning using sample re-weighting (*AL w. VAEACGAN using RW*) produces better classification results compared to Bayesian generative active deep learning without sample re-weighting (*AL w. VAEACGAN*



(d) Imbalanced MNIST (e) Imbalanced CIFAR-10 (f) Imbalanced SVHN

Figure 5.5: Training and classification results produced by our proposed Bayesian generative active deep learning with and without sample re-weighting (AL w. VAEACGAN using/without RW), active learning with ACGAN with and without sample re-weighting (AL w. ACGAN using/without RW), active learning without data augmentation, but using sample re-weighting (AL using RW), active learning without data augmentation and sample re-weighting (AL), and sample re-weighting (RW). The graphs are organised as Lenet [62] (row 1), Resnet18 [36] (row 2), and Resnet18pa [37] (row 3) on MNIST [61] (column 1), CIFAR-10 [53] (column 2), and SVHN [72] (column 3)

without RW). The advantage of using sample re-weighting is also depicted in the comparison between *AL w. ACGAN using RW* and *AL w. ACGAN without RW*, and between (*AL using RW*) and the original active learning (*AL*). Moreover, Fig. 5.5 indicates that the *AL w. VAEACGAN* [98] is superior to *AL w. ACGAN* for both the original and the combination with RW. Fig. 5.5 also shows that the usage of the Bayesian data augmentation [99] seems to be crucial due to the better classification performance of *AL w. ACGAN without RW* compared to active learning using sample re-weighting without data augmentation (*AL using RW*). Results in Fig. 5.5 also shows the importance of active learning as *AL using RW* produces better classification accuracy compared to sample re-weighting (*RW*) at three points (using the same training percentage).

In this paper, we provide more comprehensive descriptions for our previously proposed approaches, the Bayesian data augmentation [99] and Bayesian generative active deep learning [98]. We also propose a novel Bayesian generative active deep learning that targets a robust learning of deep models for imbalanced data. This approach has been shown to be more effective than Bayesian generative active deep learning on several imbalanced data sets. In the future, we plan to improve the computational complexity of the Bayesian generative active deep learning [98].

Acknowledgements

We gratefully acknowledge the support by Vietnam International Education Development (VIED), Australian Research Council through grants DP180103232, CE140100016 and FL130100102.

CHAPTER 6

Conclusion and Future Works

In this thesis, we investigated several learning approaches that aim to improve the efficiency in the use of labelled data sets for training deep models. In particular, we propose novel effective learning methods that enable deep learning models to generalise well using not only relatively small, but also imbalanced labelled training data sets. These proposed methods are: Bayesian data augmentation [99], Bayesian generative active deep learning [98], and a novel extension of the Bayesian generative active deep learning that is robust to class imbalanced data. In this chapter, we first provide in Sec. 6.1 a summary of the contributions of this thesis. We then analyse several limitations of the current work and discuss some possible future directions for this research in Sec. 6.2.

6.1 Summary of Contributions

Firstly, in Chapter 3, we propose a novel theoretically sound Bayesian data augmentation (BDA) that aims to train a generative model to produce new synthetic samples, targeting the training of more accurate deep classification models. We show that BDA is different from the current dominant data augmentation technique, which is referred to as "poor man's" data augmentation (PMDA), where the generation of artificial data points is executed only once, and prior to the training process. In our proposed BDA, the generator and the classifier are jointly trained, allowing the generator to adapt to the training process. We formulate that BDA based on a variant of the expectation maximisation (EM) algorithm, called generalised Monte-Carlo expectation maximisation (GMCEM). We provide a theoretical justification for the weak convergence of the BDA framework. To introduce a demonstration for BDA, we adapt an extension of the generative adversarial network (GAN), namely ACGAN [74], by using the generator and splitting the classifier from the discriminator. This separation is critical since it allows us to test the BDA with different sophisticated classifiers to improve state-of-the-art classification performance. We empirically show that our proposed BDA produces more accurate classification results than the PMDA and also the ACGAN model.

Secondly, motivated by the fact that, in the BDA, the generated samples are likely not informative, particularly at latter stages of the training process [98], we introduce in Chapter 4 the Bayesian generative active deep learning that targets the generation of novel synthetic data points that are informative for the training process. To formulate that Bayesian generative active deep learning, we propose a theoretically sound combination of the BDA and the Bayesian active learning by disagreement (BALD) [26, 42], in which the most informative samples are selected by BALD. These informative training instances are then used in the data generation procedure of the BDA to produce novel synthetic samples. We also theoretically show the informativeness of the generated samples. We provide empirical demonstration that shows that our proposed Bayesian generative active deep learning is superior to BDA and BALD regarding both training data efficiency and classification results.

Finally, in an attempt to facilitate the Bayesian generative active deep learning, explained in Chapter 4, to perform well on class imbalanced data sets, we strengthen that method in Chapter 5 with a novel extension that is robust to imbalanced data. This idea is realised by combining the Bayesian generative active deep learning algorithm with one of the most effective imbalanced learning methods, namely the meta sample re-weighting approach [78]. In particular, that sample re-weighting procedure is used at each iteration of the Bayesian generative active deep learning to re-balance the newly updated labelled training set. We also empirically demonstrate that our novel proposed method produces better classification results compared to the original Bayesian generative active deep learning across several imbalanced data sets and different deep classifiers.

6.2 Thesis Limitations and Future Work

In Chapter 3, we investigated the Bayesian data augmentation (BDA) that aims to train a generative model to produce new synthetic samples to avoid over-fitting in training deep models. Aside from the informativeness of the generated samples that has been addressed in the Bayesian generative active deep learning [98] in Chapter 4, another crucial factor that could be taken into account is the difficulty to fit those novel synthetic samples. This difficulty can be alleviated by the use of training samples with an increasing difficulty level that can boost the convergence of the BDA method [4,55]. One reasonable future direction is therefore to integrate curriculum training (CL) [4], or self-paced learning (SPL) [45,55] into BDA. Moreover, although BDA has been shown to be theoretically and empirically superior to PMDA, it would be interesting to introduce an efficient combination between BDA and the geometrical transformations of PMDA to utilise the "label-preserving" property of those transformations. This idea is inspired by the parameter expansion data augmentation (PX-DA) algorithm [66], in which some "distribution preserving" linear transformations are applied to latent variables to accelerate the convergence of the data augmentation using latent variables method [94]. Another potential limitation of BDA is that it can waste computational resources for training since the number of synthetic samples are the same at each iteration, while the generated samples at the initial training stages may be unrealistic or unhelpful for the training process. One possible future work to improve the running time of BDA is to design, for example, an adaptive DA, where the first training iteration would start with a small number of synthetic samples, and this value is adaptively increased after each iteration as training progresses.

Although the ability to perform well on imbalanced data set of Bayesian generative active deep learning [98] in Chapter 4 has been improved in its robust extension in Chapter 5, there have been several limitations of the Bayesian generative active deep learning that need to be addressed. Firstly, in that Bayesian generative active deep learning, the active sample selection procedure is based on the BALD acquisition function, but BALD is currently not the state-of-the-art active learning method. One possible future work is to introduce the use of some more recent approaches, for example the BatchBALD [50] in the Bayesian generative active deep learning framework. This research direction can potentially improve data diversity of the selected mini-batch at each iteration of the Bayesian generative active deep learning, and therefore may accelerate its convergence rate [50]. Another limitation of the Bayesian generative active deep learning is that, in the experiments, the stopping condition was empirically selected such that the algorithm is executed until the classification performance reaches the upper bound defined by the 10× BDA [98]. It is, therefore, an interesting future work to theoretically investigate a general stopping criterion for that algorithm. One reasonable solution may be to provide an estimate of the sample complexity [32, 85]–the algorithm will be terminated when the number of training samples reaches that sample complexity value to guarantee a given classification performance level. Furthermore, the Bayesian generative active deep learning relies on the assumption about the existence of a large unlabelled pool data, and a human annotator to label the selected informative samples, but this labelling process is expensive and prone to errors. One possible avenue for future research is to directly generate informative labelled samples for the training process without using an oracle. This can be done, for example, by integrating the sophisticated acquisition functions in the objective function of GAN training to directly generate informative samples for the training process–that is motivated by the ActiveGAN approach [51].

In Chapter 5, we propose a novel Bayesian generative active deep learning that is robust to imbalanced data. Although the empirical evidence in Chapter 5 shows that the proposed method produces better classification results than the original baseline methods on several imbalanced data sets, our approach is still relatively straightforward since it is based on a combination of two existing methods. It is an interesting future work to introduce a theoretically sound and more efficient robust Bayesian generative active deep learning by, for example, incorporating the weighted loss from the sample re-weighting scheme [78] into the objective function of the VAE-ACGAN training, or to the acquisition function of BALD. Improving the computational complexity of both the Bayesian generative active deep learning and its novel robust extension is also an interesting research direction for future work.

Bibliography

[1] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka. An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969, 1993.

[2] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

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

[4] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

[5] C. Bishop. Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn. *Springer, New York*, 2007.

[6] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[7] M. A. Carreira-Perpinan and G. E. Hinton. On contrastive divergence learning. In *AISTATS*, volume 10, pages 33–40. Citeseer, 2005.

[8] N. V. Chawla, N. Japkowicz, and A. Kotcz. Special issue on learning from imbalanced data sets. *ACM Sigkdd Explorations Newsletter*, 6(1):1–6, 2004.

[9] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2722–2730, 2015.

[10] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2016.

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

[12] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[13] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[14] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[15] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[16] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[17] X. Cui, V. Goel, and B. Kingsbury. Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(9):1469–1477, 2015.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009*, 2009.

[20] E. L. Denton, S. Chintala, a. szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems 28*, pages 1486–1494. 2015.

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

[22] S. Ertekin, J. Huang, and C. L. Giles. Adaptive resampling with active learning. *Under Review*, 2009.

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

[24] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard. Adaptive data augmentation for image classification. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3688–3692. IEEE, 2016.

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

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

[27] D. Gissin and S. Shalev-Shwartz. Discriminative active learning. 2018.

[28] I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[29] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

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

[31] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[32] S. Hanneke. Theoretical foundations of active learning. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA MACHINE LEARNING DEPT, 2009.

[33] S. Hauberg, O. Freifeld, A. B. L. Larsen, J. Fisher, and L. Hansen. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In *Artificial Intelligence and Statistics*, pages 342–350, 2016.

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

[35] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

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

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

[38] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[39] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

[40] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[41] Q. Hoang, T. D. Nguyen, T. Le, and D. Phung. Mgan: Training generative adversarial nets with multiple generators. 2018.

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

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

[44] J. Janai, F. Güney, A. Behl, and A. Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *arXiv preprint arXiv:1704.05519*, 2017.

[45] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann. Self-paced curriculum learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[46] J. M. Johnson and T. M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

[47] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017.

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

[49] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[50] A. Kirsch, J. van Amersfoort, and Y. Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *arXiv preprint arXiv:1906.08158*, 2019.

[51] Q. Kong, B. Tong, M. Klinkigt, Y. Watanabe, N. Akira, and T. Murakami. Active generative adversarial network for image classification. 2019.

[52] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

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

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

[55] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.

[56] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[57] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[58] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR.

[59] A. Lavecchia. Deep learning in drug discovery: opportunities, challenges and future prospects. *Drug discovery today*, 2019.

[60] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

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

[62] Y. LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20:5, 2015.

[63] C. Li, K. Xu, J. Zhu, and B. Zhang. Triple generative adversarial nets. *CoRR*, abs/1703.02291, 2017.

[64] S. Lim, I. Kim, T. Kim, C. Kim, and S. Kim. Fast autoaugment. *arXiv preprint arXiv:1905.00397*, 2019.

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

[66] J. S. Liu and Y. N. Wu. Parameter expansion for data augmentation. *Journal of the American Statistical Association*, 94(448):1264–1274, 1999.

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

[68] D. Masko and P. Hensman. The impact of imbalanced training data for convolutional neural networks, 2015.

[69] E. G. Miller, N. E. Matsakis, and P. A. Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471. IEEE, 2000.

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

[71] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

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

[73] A. Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

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

[75] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *bmvc*, volume 1, page 6, 2015.

[76] W. Qian and D. Titterington. Estimation of parameters in hidden markov models. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 337(1647):407–428, 1991.

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

[78] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pages 4331–4340, 2018.

[79] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

[80] M. B. Ring. *Continual learning in reinforcement environments*. PhD thesis, University of Texas at Austin Austin, Texas 78712, 1994.

[81] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[82] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[83] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[84] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

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

[86] B. Settles, M. Craven, and L. Friedland. Active learning with real annotation costs. In *Proceedings of the NIPS workshop on cost-sensitive learning*, pages 1–10. Vancouver, CA, 2008.

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

[88] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[89] P. Y. Simard, D. Steinkraus, and J. C. Platt. 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.

[90] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[91] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3308–3318, 2017.

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

[93] R. S. Sutton, A. G. Barto, et al. *Introduction to reinforcement learning*, volume 2. MIT press Cambridge, 1998.

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

[95] M. A. Tanner. Tools for statistical inference: Observed data and data augmentation methods. *Lecture Notes in Statistics*, 67, 1991.

[96] M. A. Tanner and W. H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540, 1987.

[97] K. M. Ting. A comparative study of cost-sensitive boosting algorithms. In *In Proceedings of the 17th International Conference on Machine Learning*. Citeseer, 2000.

[98] T. Tran, T.-T. Do, I. Reid, and G. Carneiro. Bayesian generative active deep learning. In *International Conference on Machine Learning*, pages 6295–6304, 2019.

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

[100] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, page 1, 2019.

[101] D. A. Van Dyk and X.-L. Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

[102] D. A. Van Dyk, X.-L. Meng, et al. Cross-fertilizing strategies for better em mountain climbing and da field exploration: A graphical guide book. *Statistical Science*, 25(4):429–449, 2010.

[103] S. Wang and X. Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):1119–1130, 2012.

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

[105] C. Yoon, G. Hamarneh, and R. Garbi. Generalizable feature learning in the presence of data bias and domain class imbalance with application to skin lesion classification.

[106] X. Zhang and Y. LeCun. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*, 2015.

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

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

[109] X. Zhu, J. Lafferty, and R. Rosenfeld. *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, language technologies institute, school of . . . , 2005.

[110] Q. Zou, S. Xie, Z. Lin, M. Wu, and Y. Ju. Finding the best classification threshold in imbalanced classification. *Big Data Research*, 5:2–8, 2016.