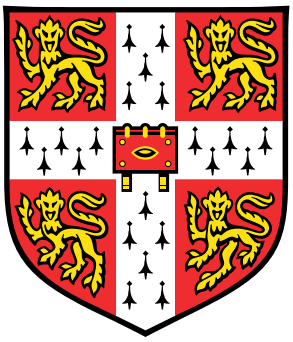
Active Learning for High Dimensional Inputs using Bayesian Convolutional Neural Networks



Riashat Islam

Department of Engineering

University of Cambridge

This dissertation is submitted for the degree of

Master of Philosophy

St John's College August 2016

I would like to dedicate this thesis to my loving parents ...

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is the result of my own work in collaboration with my assigned supervisors, except where specifically indicated in the text. This dissertation contains less than 14,000 words including appendices, bibliography, footnotes, tables and equations and has less than 150 figures.

Riashat Islam
August 2016

Acknowledgements

I would sincerely like to thank my supervisors, Zoubin Ghahramani and Yarin Gal for their expert advice and support, whilst also giving me the freedom to work on things of my interest. They have been tremendously supportive and have guided my work to the fullest. Zoubin has been an unswerving source of inspiration for me. His knowledgeable advice helped me to explore exhilarating areas of machine learning, and helped me work on a project of my interest. I would also like to give special thanks to Yarin Gal, without whose ideas, support and patience, I would not have this project see this day. Thank you Yarin for suggesting me to work in this direction, for giving me your valuable time while conducting our discussions and for fuelling my avid curiosity in this sphere of machine learning.

I would also like to thank other members in the Computational and Biological Learning Lab, Machine Learning Group at University of Cambridge for their enormous support. Special thanks goes to Richard Turner who provided me useful advice throughout my time at Cambridge. I would also like to thank Shane Gu, Yingzhen Li, Thang Bui and Matt Hoffmann for their advice and support throughout my degree. I am lucky to meet Vera and Ambrish through the course of this MPhil degree.

I am grateful to St John's College for the graduate access studentship scheme. I sincerely thank the Cambridge International Trust and Commonwealth Scholarship and Fellowship Program for awarding me the Cambridge Assessment Scholarship, which made my coming to Cambridge a reality.

I would like to thank my parents Siraj and Shameem - the most important people in my life who have always put my well-being and academic interests over everything else, I owe you two everything. My life here in UK would not have felt like home without the amazing relatives that I have here - thank you, Salma, Rafsan, Tisha and Amirul for always being there and supporting me through all these years of my living in UK. I would like to thank Tasnova for her amazing support and care, and for bringing joy and balance to my life. Thanks to my friends Rashik, Riyasat, Raihan, Mustafa, Mahir, Sadat and Imtiaz for standing by my side over the long years.

I am always thankful to Almighty Allah for the opportunities and successes He has given me in this life, I would not be what I am today without His blessings.

Abstract

The recent success of deep learning in applied machine learning gained tremendous success, addressing the problem of learning from massive amounts of data. However, the challenge now is to learn data-efficiently with the ability to learn in complex domains with scalable practical applications without requiring deep learning models to be trained with large quantities of data. We present the novel framework of achieving data-efficiency in deep learning through active learning. Since labelled data for deep learning is costly to collect, here we develop active learning algorithms for collecting the most informative data for training deep neural network models. Our work is the first to propose active learning algorithms for image data using convolutional neural networks.

Recent work showed that the Bayesian approach to CNNs can offer robustness of these models to overfitting on small datasets. By using dropout in neural networks to avoid overfitting as a Bayesian approximation, we can represent model uncertainty from CNNs for image classification tasks. Our proposed Bayesian active learning algorithms use the predictive distribution from the output of a CNN to query most useful datapoints for image classification with least amount of training data. We present information theoretic acquisition functions which incorporates model uncertainty information, namely Dropout Bayesian Active Learning by Disagreement (Dropout BALD), along with several new acquisition functions, and demonstrate their performance on image classification tasks using MNIST as an example. While our approach is the first to propose active learning in a deep learning framework, we compare our results with several semi-supervised learning methods which also focuses on learning data-efficiently using least number of training samples.

Our results demonstrate that we can perform active learning in a deep learning framework which has previously not been done. This allows us to achieve data-efficiency in training. We illustrate that compared to standard semi-supervised learning methods, we achieve a considerable improvement in classification accuracy. Using our Bayesian active learning framework we achieve classification accuracy which is close to the currently published state of the art results for MNIST dataset.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

This thesis introduces for the first time a Bayesian active learning framework for high dimensional inputs (such as images) for use in Deep Learning through the use of Bayesian Convolutional Neural Networks. It proposes an active learning approach towards data-efficient deep learning. We take a probabilistic Bayesian approach for information theoretic active learning by representing model uncertainty in deep learning for image classification tasks using Bayesian convolutional neural networks.

In chapter 1, we give a brief introduction to Bayesian active learning and how to capture model uncertainty in deep learning for classification tasks. We build on a tool that casts dropout training in neural networks as approximate Bayesian inference. In chapter 2, we will demonstrate how to use to propose an information thereotic entropy based active learning framework based on Bayesian CNNs in chapter 2. We propose several new acquisition functions which incorporates uncertainty information for active learning in image classification tasks, and demonstrate the novelty of our work. Chapter 3 provides the experimental results illustrating the performance of our Bayesian active learning algorithms with dropout uncertainty from Bayesian CNNs. We note that our approach is the first to propose active learning for image data based on deep learning tools such as CNNs, which is achievable by considering approximate Bayesian inference which provides robustness to over-fitting on small datasets. Finally, chapter 4 discusses and summarises the results, and includes possible future work. We provide state-of-the-art performance for image classification task, and introduce novel Bayesian active learning frameworks that can be used in deep learning to achieve data-efficiency.

1.1 Data-Efficient Machine Learning

Recent approaches in machine learning are focused on learning from massive amounts of data. Deep learning approaches have been shown to provide highly scalable solutions. In applications such as image and speech recognition, machine translation, speech synthesis and recommendation systems, deep neural networks have achieved state of the art performance when trained with large amounts of training data [1, 2]. Convolutional neural networks in deep learning have been shown to achieve state of the art performances in image processing tasks [3]. However, CNNs are known to require large amounts of training data, and can quickly overfit when trained with small datasets. Training with large datasets also often require enormous computational resources and hence training these deep neural network models can become difficult. While Bayesian neural networks are robust to overfitting and can be trained with small datasets [4, 5], their CNN counterparts could not be attempted successfully due to the problem of modelling the distribution over kernels in the CNN. Recently, however, the use of efficient Bayesian CNNs have been shown which can offer better robustness to overfitting on small datasets [6].

Data-efficiency has become an increasingly important requirement for modern machine learning and artificial intelligence systems. The task of data-efficient machine learning is to ask how can we design efficient machine learning systems that can learn using the least amount of data while also achieving similar levels of performance and providing scalable solutions. This is especially important in domains such as personalized healthcare, robotic systems and reinforcement learning since data is scarce in such domains. It is important to be able to learn data-efficiently in these small data domains. In this work, we therefore demonstrate the ability to learn in a complex domain without requiring large quantities of data. We focus on the task of training a deep learning model with the least amount of training data through the use of a Bayesian active learning framework.

1.2 Introduction to Bayesian Active Learning

In active learning, the goal is to produce the best machine learning model with the least amount of training data. The learner in active learning seeks the most informative data to train the model upon. This is particularly useful since there is vast amount of unlabelled data that is available to us, but it is often costly to obtain labels for all the data. Active learning algorithms therefore seek the most useful data for training sets in machine learning [7]. Active learning algorithms are particularly of importance in computer vision tasks where

it is time and cost consuming to obtain a good set of labeled images. Building robust image classifiers requires large number of labelled training data instances. In this work, we aim to develop an efficient active learning method to build a competitive classifier with a limited amount of labelled training instances. However, training a good classifier with minimal labeling cost is a critical challenge posed in machine learning research. We focus on the pool based active learning setting by evaluating the informativeness of instances with the most uncertainty measures which assumes that an instance with a higher classification uncertainty is most critical to the label. We propose several active learning query strategies which uses the uncertainty estimates obtained in a deep learning setting.

We consider using the Bayesian framework for active learning which can be used for the design of active learning algorithms considering an information theoretic approach [8]. Within a Bayesian active learning framework, acquisition functions can be used that can measure the expected informativeness of pool points from which to actively select the next data point to be added to training set. In this work, we take the information theoretic approach to probabilistic active learning, where the acquisition functions can measure the utility of a datapoint by quantifying its informativeness about the parameters. By using a well calibrated uncertainty estimate from the predictions made by the model, which is briefly introduced later in section 1.3, we introduce our Bayesian active learning framework called Dropout Bayesian Active Learning by Disagreement. Later in chapter 2 we discuss the properties of these acquisition functions and its reliance on using a good uncertainty estimate obtained from using a Bayesian convolutional neural network.

1.3 Representing Model Uncertainty in Deep Learning

Recent work showed how model uncertainty can be captured in deep learning by taking a Bayesian approach to dropout in neural networks (NNs) [9]. By considering the relation between Gaussian Processes (GPs) and dropout for regularisation in NNs, it has been shown that uncertainty can be obtained in deep learning classification and regression tasks. We build our work on this framework to use the uncertainty information in image classification tasks for active learning. This is particularly useful since the model can now classify images in CNNs with certain confidence, and we can use active learning to treat the inputs that the model is uncertain about. Inputs to the CNN that the model is highly uncertain about can now be queried in pool-based active learning setting, and passed onto the active learner for obtaining the correct label. [9] showed that a neural network with arbitrary depth and non-linearity, with dropout applied after every weight layer is equivalent to an approximation

to the probabilistic deep Gaussian process [10].

The Bayesian approach to dropout in NNs have also been extended for use in CNNs. By placing a distribution over the kernels (Gaussian filters) of a CNN model, [6] showed that we can approximate the CNN model's intractable posterior with Bernoulli variational distributions. [6] proposed practical dropout CNN architectures, the Bayesian CNN model and showed that these models can reduce overfitting on small datasets. By performing dropout after every convolutional layer at training, and by evaluating the model output by approximating the posterior with average stochastic forward passes through the model at test time, we can capture model predictive uncertainty. Note that the uncertainty obtained from this approach is significantly different to the probabilities obtained from a softmax classifier. A softmax function only approximates the relative probabilities between the class labels and do not provide an overall measure of the model's uncertainty. By considering model uncertainty in deep learning, we propose novel information theoretic active learning algorithms that relies on the uncertainty calibrations obtained from this approach. The predictive uncertainty from Bayesian CNN models in image data shows the image pool set points that the model is uncertain about. This uncertainty is then used for our proposed acquisition functions for Bayesian active learning.

1.4 Active Learning in Deep Learning framework

In this work, we specifically focus on active learning in a deep learning framework for image datasets. While active learning has been well known in the machine learning research community for a long time, these settings are not typically used with deep learning systems. This is because deep neural networks require large amounts of training data for training. Furthermore, convolutional neural networks which are typically used for image classification are known to be highly prone to overfitting when trained with small datasets. For this reason, CNNs had not been previously used in an active learning setting for images.

The Bayesian convolutional neural network (Bayesian CNN) approach uses the Bayesian framework for overfitting similar to other Bayesian probabilistic methods, by casting dropout training in neural networks as approximate Bayesian inference. Furthermore, these models, similar to other Bayesian frameworks such as Gaussian Processes, can also be used to represent uncertainty for classification tasks. While Gaussian Processes are known to offer good uncertainty estimates for regression, and more recently with classification, GPs are known not to be quite robust in providing uncertainty estimates for high dimensional

inputs, especially in classification tasks. Bayesian ConvNets on the other hand have been shown to work quite well for classification tasks, offering good uncertainty estimates. By combining these ideas of avoiding overfitting on small data and using model uncertainty from Bayesian ConvNet, we introduce the framework of Bayesian active learning in deep learning. We present several novel active learning acquisition functions which uses a Bayesian CNN architecture, and compare our proposed methods with several other approaches but using the traditional CNN model which in contrast, are prone to over-fitting with small datasets.

We emphasize that this is the first step towards using active learning based on the use of CNNs in a deep learning framework. By considering Bayesian approach to CNNs, achieving robustness to overfitting on small datasets and obtaining Bayesian model uncertainty, we show that active learning can also be used in a deep learning setting for image classification tasks towards the goal of achieving data-efficiency.

Chapter 2

Bayesian Active Learning in Deep Learning

In this chapter, we introduce the Bayesian framework of representing model uncertainty in deep learning to design our information theoretic active learning algorithms. In section 2.1 we briefly introduce the Bayesian information theoretic approach to active learning, and then describe the use of model predictive uncertainty in deep learning for our acquisition functions in section 2.2. In section 2.3 we describe and introduce our proposed acquisition functions that can be used for image data using Bayesian CNN models. We discuss that these acquisition functions are mainly based on being able to represent model uncertainty from a deep learning model. In section 2.4.1 we discuss several related work which focuses on modelling uncertainty in deep learning, and demonstrate how our proposed methods are suitable, easy to compute and extendable for CNNs compared to other methods in an active learning setting, especially considering high dimensional inputs such as images.

2.1 Information Theoretic Active Learning

Active learning algorithms focus on selecting their own training data for training machine learning models. Active learning can be performed in three scenarios such as *continuous sampling*, *pool based* and *stream based* active learning. We consider the task of pool-based active learning in which the learner has access to a pool of unlabelled data from which to select points for annotation. In order to select the most informative points that the learner must choose for the training data, active learning algorithms must assign a score or utility to each location in the input space that can be queried. This utility function is evaluated for every point in the pool set. Such utility functions can be built using an information

theoretic approach. Information theoretic approaches are broadly under the category of probabilistic active learning. Pool based active learning have many applications including text classification [11], image classification [12], speech recognition [13] and recommendation systems [14]. Within the Bayesian active learning framework, utility or acquisition functions can measure the expected informativeness of candidate measurements.

2.1.1 Information Theory

We first give a brief overview to information theory before presenting our information-theoretic active learning approach. Information theory was founded by Claude Shannon [15] where he derived a theoretic upper bound to the capacity of a channel, which is the maximum rate that a set of symbols can be transmitted with zero reconstruction error. The information content of a datapoint x and the entropy which is the average information content in the ensemble is given by:

$$J(x) = -\log P(x) \tag{2.1}$$

$$H[P(x)] = -\sum_{x} P(x) \log P(x) \tag{2.2}$$

where $J(x)$ measures the information content of a data point x, and $H[P(x)]$ is the entropy. Entropy is a measure of the uncertainty in a distribution.

Two other information theoretic quantities that occur frequently in machine learning are the *mutual information* and *Kullback-Leibler* (KL) divergence. The mutual information between two random variables X and Y is given by:

$$I[X,Y] = H[p(X)] - \mathbb{E}_{p(Y)} H[p(X|Y)] \tag{2.3}$$

where $\mathbb{E}_{p(Y)} H[p(X|Y)]$ is the conditional entropy denoted by $H(X|Y)$. It is also symmetric and measures how much information X carries about Y and vice versa. Shannon showed that the maximum capacity of a channel is given by the mutual information between the sent and received signals. The KL divergence which is a measure of dissimilarity between two probability distributions p(X) and q(X), has the intuition as the number of additional bits needed to transmit symbols with distribution p(X), if our model of the distribution is q(X).

2.1.2 Information Gain Utility Functions

In pool based active learning, each labelled training example belongs to a certain class that is denoted by $y \in 1, ...k$. However, we do not know the true class labels for the examples in the active pool. We consider entropy which is a measure of uncertainty of a random variable. Entropy values can indicate the class membership of the predicted labels Y where the higher values of entropy can imply more uncertainty in the distribution. In other words, this means that if an example unlabelled point in the pool set has a distribution with a higher entropy, then the classifier is more uncertainty about its class membership.

Equation 2.2 is a measure to quantify uncertainty in a probability distribution. In Bayesian active learning, the goal is to query points from a pool set such as to minimize the posterior entropy after collecting data. The points are queried based on the expected information gain which is given by:

$$U(x) = H[p(\theta|D)] - \mathbb{E}_{p(y|x,D)}H[p(\theta|D,x,y)] \tag{2.4}$$

Equation 2.4 is equivalent to the mutual information between the parameters and the unobserved output, conditioned upon the input and the observed data.
Equation 2.4 was first proposed for the design of Bayesian experiments in [16]. However, equation is difficult to compute due to the intractability of the Bayes rule and therefore mathematical approximations are usually required when using equation 2.4 for complex models. Another perspective to consider for information theoretic active learning is based on maximizing the KL divergence between the current posterior and the next posterior such that $KL[p(\theta|D,x,y)||p(\theta|D)]$.

In our work, we propose an active learning acquisition functions based on the equivalent formulation of equation 2.4 that was initially proposed in [17] called Bayesian Active Learning by Disagreement (BALD). As discussed later in section 2.9, [17] showed that the different formulation of equation 2.4 can provide substantial practival advantages for computation. Later in section 2.9, we propose our Dropout BALD acquisition function which combines model uncertainty with the expected information gain for our proposed acquisition function.

2.2 Bayesian Convolutional Neural Networks

In this section, we briefly introduce the model uncertainty framework for deep learning that was introduced in [6, 9]. Recent work in [6, 9] have shown that deep learning techniques

can be used to reason about uncertainty over the features by using a Bayesian approach to dropout training in neural networks. [9] have shown that a Bayesian approximation to dropout training can be used to capture the confidence of the model in its prediction. Dropout applied after every weight layer is mathematically equivalent to the well known Bayesian model, the Gaussian Process. The Bayesian approach to dropout training makes these deep learning models more robust to over-fitting as Bayesian frameworks have already been shown to be robust to overfitting. In addition, such frameworks can provide an interpretation to reason about uncertainty in deep learning and allows the introduction of the Bayesian machinery in existing deep learning frameworks. Standard deep learning models used for classifiaction tasks cannot capture the model uncertainty, and the softmax output of such models are often misinterpreted as the model confidence. The softmax output of a deep model does not necessarily quantify how uncertain the model is about its predictions.

[9] uses Bayesian probability theory to offer a ground tool to reason about uncertainty, and have showed that the use of dropout in NNs can be interpreted as a Bayesian approximation of a well known probabilistic model, the Gaussian Processes. While Dropout is commonly used in deep learning as a way to avoid overfitting, [9] interpretation suggests that dropout approximately integrates over the model's weights, and the mathematical similarly between Gaussian Processes and dropout can be used to develop a tool that can represent uncertainty in deep learning.

Based on [9], the use of dropout in NNs was further used for proposing Bayesian CNN architectures in [6]. Previously, Bayesian CNNs could not be implemented due to the difficulty of inferring the model posterior when having a large number of parameters. Even with small number of parameters, inferring the model posterior in a Bayesian NN was a difficult task since variational inference based on the use of Gaussians for variational distribution to approximate the posterior was computationally expensive. For example, using a Gaussian approximating distribution to model the posterior to be close to the true posterior increases the number of model parameters significantly. Therefore, such approaches could not previously be used for CNNs since the increase in number of parameters in CNN architectures can be more expensive. However, recently, [6] showed that by using a Bernoulli approximating variational distribution, we can approximate the posterior with no additional parameters which led to the efficient implementation of Bayesian CNNs.

[6] proposed dropout CNN architectures showing that dropout networks training can be cast as approximate Bernoulli variational inference, and that the implementation of Bayesian

CNN is simply performing dropout after every convolution layer at training. Furthermore, by performing dropout at test time, [6] showed that Bayesian CNN models can be implemented very efficiently, and can be used to evaluate the model output by approximating the predictive posterior. The implementation of Bayesian CNNis therefore simply using dropout after every convolution layer before pooling. At test time, by performing several average stochastic forward passes through the model, ie, referred to as Monte-Carlo (MC) dropout, the approximating predictive posterior can be easily obtained. This also means that by performing MC dropout at test time, ie, using averaging stochastic forward passes through the model at test time, we can approximate the predictive distribution. This in other words gives us a measure of uncertainty over the classification predictive probabilities obtained from the Bayesian CNN MC dropout architectures.

By using these uncertainty estimates from the predictive distribution of a Bayesian CNN model, we develop our information theoretic approach to active learning. In other words, the uncertainty over predictions, the predicted probabilities can further be used to measure entropy, which can quantify uncertainty for the active learning algorithm. We propose several new active learning acquisition functions based on utilizing these MC dropout uncertainty estimates and a Bayesian CNN classifier such as to derive a data-efficient active learning framework for image classification tasks in deep learning.

2.3 Active Learning Acquisition Functions

In this section, we introduce our proposed active learning acquisition functions which uses Monte-Carlo (MC) dropout to obtain a predictive distribution from a Bayesian CNN architecture. Our proposed acquisition functions uses the approximating predictive distribution as a measure of uncertainty to compute our acquisition functions $U(x)$. First, we describe our active learning setting as below.

We consider only the pool-based active learning setting for active learning of high dimensional inputs such as images. Suppose we have a set of N images with each image belonging to one of the L possible classes. We divide the training set into train, validation and pool set, and we assume that the class labels for images in the pool set are unknown. The active learner has access to a set or pool of unlabelled data from which to select points for annotation. According to an acquisition function, the active learner chooses one or more of the N images, and these images are presented to the oracle that can provide the correct class labels. The active learner chooses additional images at each round in the algorithm from the unlabelled

set that would be particularly informative if their labels were known.

More formally, let U^t be the pool of unlabelled images at the start of round t and let L^t be the corresponding pool of labelled images. We assume that the acquisition function queries 1 or more images at each round according to a given accquisition function to choose the most informative query point. This process leads to new labelled and unlabelled sets for the next round.

$$L^{t+1} = L^t \cup x^t, y^t \tag{2.5}$$

$$U^{t+1} = U^t \backslash x^t \tag{2.6}$$

where $x^t \in$U^t is the example chosen in round t and y^t is its label assigned by the oracle. In pool-based active learning, the acquisition functions evaluates the pool points and ranks the entire collection of pool points from which the best queries are selected. Below, we describe each of our acquisition functions.

Note that all our active learning algorithms are based on Bayesian CNNs for image classification tasks. The predicted probabilities are obtained from the softmax output of a CNN and model uncertainty is obtained by using test time MC dropout. Based on these, we construct our acquisition functions for query selection as described in the sections below. Later, in chapter 3, we will provide the experimental results using each of our acquisition functions, and demonstrate their effectiveness.

2.3.1 Dropout Bayesian Active Learning by Disagreement

We consider an information theoretic Bayesian active learning setting using entropy to quantify the uncertainty from the predictive probability distribution, which is the natural objective to minimize the posterior entropy after collecting data. Following the approach taken by [17], we consider taking a myopic greedy approach, selecting the next pool point as if it were the last. The acquisition function developed by [17], shows that the expected

information gain was equivalent to the mutual information between the parameters and the observed output as follows:

$$\begin{aligned} U(x) &= H[p(\theta|D)] - \mathbb{E}_{p(y|x,D)} H[p(\theta|D,x,y)] \\ &= I[\theta, y|D,x] \\ &= H[p(y|x,D)] - \mathbb{E}_{p(\theta|D)} H[p(y|x,\theta)] \end{aligned} \tag{2.7}$$

Equation 2.7 shows the acquisition function known as the Bayesian Active Learning by Disagreement (BALD), which provides the intuition that the first term seeks the input **x** for which the model has high uncertainty about the output **y** and the second term seeks a datapoint with low expected conditional uncertainty $\mathbb{E}_{p(\theta|D)} H[p(y|x,\theta)]$. In other words, this acquisition function will reward data points whose output has high entropy due to parameter uncertainty, which is captured by the marginal predictive distribution $p(y|x,D)$, but penalizes uncertainty due to inherent noise which is modelled by the likelihood $p(y|x,\theta)$.

Equation 2.7 can be approximated using Monte Carlo samples from the posterior. $U(x)$ can be estimated using samples using BALD as follows:

$$U(x) \approx H[\frac{1}{N}\sum_{i=1}^{N} p(y|x,\theta_i)] - \frac{1}{N}\sum_{i=1}^{N} H[p(y|x,\theta_i)] \tag{2.8}$$

Following equation 2.8, we derive dropout BALD, which uses the Monte-Carlo samples of the predicted distribution obtained from using test-time dropout of the Bayesain CNN implementation. For obtaining the predicted class probabilities $p(y|x)$, we use the Bayesian CNN implementation with dropout used after every parameter layer. We average T stochastic forward passes through the model following the Bayesian interpretation of CNNs and obtain MC dropout samples of predicted class probabilities. The approach of using dropout at test time is by Monte Carlo averaging of stochastic forward passes through the model. The MC dropout testing applied to CNNs gives us noisy estimates with potentially different test results over different runs. Using this, we can therefore construct our Dropout BALD acquisition function as follows, where k is the number of Monte-Carlo approximations used for the predictive probabilitiy distribution from a Bayesian CNN output.

$$U(x) \approx H[\frac{1}{k}\sum_{i=1}^{k} p(y_i|x_i)] - \frac{1}{k}\sum_{i=1}^{k} H[p(y_i|x_i)] \tag{2.9}$$

Equation 2.9 shows the Dropout Bald acquisition function based on the expected information gain for choosing the best query points from the pool set. $U(x)$ queries points which maxi-

mizes the expected information again $x^* = \arg\max U(x)$. The acquisition function can be interpreted as follows: $H[\frac{1}{k}\sum_{i=1}^{k} p(y_i|x_i)]$ is the entropy of the average predicted probability, ie, the learner seeks points about which the model is marginally most uncertain about the average predicted probability. The learner seeks the point about which the model is most uncertain about the average output. The second term in $U(x)$ given by $\frac{1}{k}\sum_{i=1}^{k} H[p(y_i|x_i)]$ seeks the point for which the average uncertainty is low. The Dropout BALD acquisition (equivalently can be called as MC Dropout Bald) function, or can therefore be interpreted as follows: the learner queries point based on the expected information gain which is given by the uncertainty of the average output minus the average uncertainty in the output.

Our proposed active learning algorithm using the Dropout BALD acquisition function is described in algorithm box 1 below.

Algorithm 1 Active learning algorithm using Dropout BALD

```
Input: Labelled Set L, Unlablled Set U
Build CNN model
for Number of queries to make from pool set do
    for Number of Monte-Carlo Dropout samples do
        Compute predicted probabilities on pool set
    end for
    Compute entropies using average predicted probabilities
    Compute U(x) using equation 2.9
    Find best query point x_k using U(x)
    Add x_k, y_k to L, remove x_k from U
    Re-train CNN model with new training set L
    Evaluate CNN output on test set
end for
```

2.3.2 Dropout Variation Ratio

We propose another variant of acquisition function based on using the model uncertainty obtained from our Bayesian CNN implementation. For each point in the pool set, for each of the MC test time dropout samples, we now compute the predicted labels, which are different in each test time dropout. Based on these different predicted labels for each point in pool set, we can then compute a histogram of the class labels predicted by the model for each pool point. By computing this histogram, we can then compute which label the model is most confident about on average.

In other words, we compute the variation ratio for each point in the pool set. Similar to the standard deviation, the variation ratio is a measure of statistical dispersion in normal distributions. By compute the histogram of predicted labels for each point, we can compute the mode label predicted by the model. The variation ratio is the proportion of cases which are not the mode. It is given by:

$$v = 1 - \frac{f_m}{N} \tag{2.10}$$

where f_m is the frequency of the number of classes of the mode label and N is the total number of MC dropout samples. Our acquisition function called Dropout Variation Ratio is therefore given by:

$$U(x) = 1 - \frac{f_m}{N} \tag{2.11}$$

and the active learner selects the points which has the highest variation ratio, ie, $x^* = \underset{x}{\text{argmax}} U(x)$. Similar to the standard deviation, the larger the variation ratio, the more differentiated or dispersed are the class predicted labels, and the smaller the variation ratio, the more concentrated and similar are the predicted labels. Since in active learning, our learner seeks the point about which the model is most uncertain about, therefore higher values of variation ratio imply more uncertainty about the predicted labels. In other words, if variation ratio is high, it implies that the model is not too confident about a particular label, but rather assigns similar proportions to all the class labels, implying that it is uncertain about all the labels and not too confident about a particular class membership.
Our proposed active learning algorithm based on computing variation ratio from MC dropout samples of predicted classes, called "Dropout Variation Ratio" is shown in algorithm box 2 below.

Algorithm 2 Active learning algorithm using Dropout Variation Ratio

```
Input: Labelled Set L, Unlablled Set U
Build CNN model
for Number of queries to make from pool set do
    for Number of Monte-Carlo Dropout samples do
        Compute predicted classes on pool set
    end for
    Construct histogram of predicted classes from MC dropout samples
    Find the mode predicted class
    Compute variation ratio U(x) using equation 2.11
    Find best query point x_k using U(x)
    Add x_k, y_k to L, remove x_k from U
    Re-train CNN model with new training set L
    Evaluate CNN output on test set
end for
```

2.3.3 Dropout Maximum Entropy

We propose another aquisition function based on the maximum entropy measure, in which query points are selected about which the model has highest uncertainty. This is similar to the usual maximum entropy based acquisition function commonly used in active learning. This is in accordance to the uncertainty sampling acquisition function commonly used, where the learner attempts to label those instances for which the model is least certain about how to label. Our entropies are calculated based on the average of the predictive probability distribution obtained from MC dropout output samples. The entropy measure for k class classification is given by:

$$E(x) = -\sum_{i=1}^{k} p_i \log(p_i) \tag{2.12}$$

And the acquisition function selects the point which has maximum information content. However, to select points based on the model uncertainty, we would need a good uncertainty calibration which we obtain using our Bayesian ConvNet implementation. By averaging T stochastic forward passes through the model, and performing dropout at test time, we again get an uncertainty estimate over our predicted labels. In other words, our "Dropout Maximum Entropy" acquisition function incorporates the model uncertainty (ie, the uncertainty in the predictions made by the model) to calculate the entropies, which in itself is a measure of uncertainty. Later in experimental results, we will show that in a Bayesian CNN framework,

this approach outperforms than simply calculated the entropies calculated from the predictive probabilities of a single pass through the model. Our proposed acquisition function is given as follows:

$$U(x) = H[\frac{1}{k}\sum_{i=1}^{k} P_i] \tag{2.13}$$

and the query points are selected which maximizes the entropy $x^* = \arg\max_x U(x)$. The entropy is computed based on the average model uncertainty about the class membership of each points in the pool set.

Our proposed active learning algorithm based on computing entropies using average predicted probabilities, called "Dropout Max Entropy" is therefore given by algorithm box 3 below.

Algorithm 3 Active learning algorithm using Dropout Maximum Entropy

Input: Labelled Set L, Unlablled Set U
Build CNN model
for Number of queries to make from pool set **do**
    for Number of Monte-Carlo Dropout samples **do**
        Compute predicted probabilities on pool set
    end for
    Compute entropy using average predicted probabilities
    Compute U(x) using equation 2.13
    Find best query point x_k using U(x)
    Add x_k, y_k to L, remove x_k from U
    Re-train CNN model with new training set L
    Evaluate CNN output on test set
end for

2.3.4 Dropout Bayes Segnet

Our next acquisition function is based on computing the sum of standard deviations for each class label for each pool point. This can be formalised as follows. For each point in the pool set, we again perform dropout at test time, and obtain an uncertainty measure over the predicted labels for each point in the pool set. In other words, considering each pool set point, our model predicts class probabilities for each of the L classes. For MC dropout samples, we can then compute the standard deviation of probabilities for each of the L classes for each pool point. Our Bayes Segnet measure then computes the sum of standard

deviation of probabilities across the L classes for each pool set point. This therefore gives us an uncertainty estimate for each pool set point, which the active learner then uses to query points with highest sum of standard deviation of probabilities. This can be given as follows:

$$U(x) = \sum_{i=1}^{L} \sigma_i \tag{2.14}$$

where L is the number of classes under the L image classification setting. Our learner then seeks pool points with the highest $U(x) = \sum_{i=1}^{L} \sigma_i$.

However, note that, unlike the variation ratio, the standard deviation of probabilities is *not a good measure of uncertainty*. This will be further justified in experimental results section, where we show the importance of a good uncertainty measure for active learning. We understand that standard deviation of probabilities is not a good measure to use for our acquisition functions. However, through this, we demonstrate the significance of obtaining a good model uncertainty estimate from MC dropout samples.

Algorithm 4 Active learning algorithm using Dropout Bayes Segnet

```
Input: Labelled Set L, Unlablled Set U
Build CNN model
for Number of queries to make from pool set do
    for Number of Monte-Carlo Dropout samples do
        Compute predicted probabilities on pool set
    end for
    Compute σ of predicted probabilities for each pool point
    Compute U(x) using equation 2.14 based on sum of σ
    Find best query point x_k using U(x)
    Add x_k, y_k to L, remove x_k from U
    Re-train CNN model with new training set L
    Evaluate CNN output on test set
end for
```

2.3.5 Other Baseline acquisition functions

Our proposed acquisition functions are mainly based on using Bayesian CNN model architectures. We note here that even though active learning had been a major research area for quite a long time, previous methods in active learning did not use CNN models, especially in a deep learning framework. As stated previously, this is mainly because most deep learning

models were previously known to require large amounts of training data, making active learning not a suitable approach.

We compare our proposed active learning methods in a deep learning framework with several other commonly used acquisition functions. While previously these methods were commonly implemented using Support Vector Machine (SVM) or other machine learning classifiers, in this work we implement these "baseline acquisition functions" using CNN models. In the sections below, we introduce these baseline acquisition functions with which we compare our proposed algorithms.

Maximum Entropy

We compare all our proposed acquisition functions with the max entropy based acquisition function in which the learner chooses query points which has the maximum entropy. Here, we simply use a CNN model instead of our Bayesian CNN implementation, and based on the computed probabilities from the softmax output of a CNN, we can compute the entropy values for each pool point. Unlike our previously introduced "Dropout Max Entropy" acquisition function, here we simply use the predicted output probability from the softmax output of a CNN, and use the predicted probability for each class for each pool point to compute the entropy for that point.

Maximum Margin : Best vs Second Best (BvSB)

Even though entropy based active learning can be considered as a better measure for query point selection, there are several drawbacks to using an entropy based approach. The entropy measures are highly influenced by the probability values of the unimportant classes. Considering a situation where the classifier estimates the probability values of two examples in a L class problem. For one example, the classifier might assign higher and almost equal probabilities to two classes, whereas for the other example, the classifier may assign a much higher probability to only one class compared to all the others. From the classification perspective, it can be argued that the classifier is more confused about the first example than the second since the first example has two close probability values to two classes, so it is more confused about the first example than the second. However, after computing entropies, the small probability values of unimportant classes will contribute to a higher entropy score even though the classifier is much confident about the classification of the example.

Based on this, we compare our acquisition functions with non entropy based approaches, and use the softmax output of a CNN to compute the class predicted probabilities. As in [18], instead of relying on the entropy score, we consider the difference between the probability values of the two classes having the highest estimated probability value as a measure of uncertainty. The acquisition function can therefore be written as:

$$U(x) = P(y_1|x) - P(y_2|x) \tag{2.15}$$

where y_1 and y_2 are the two most probable values. This is referred to as the Best-versus-Second-Best (BvSB) approach, and the learner queries the point which has the minimum difference, ie, $x^* = \arg\min_x P(y_1|x) - P(y_2|x)$. Such a measure is a more direct way of estimating confusion about class membership from a classification standpoint.

Least Confident

This is a baseline uncertainty sampling based least confident measure in which the active learner chooses query points about which the model is least confident about. This baseline acquisition function computes the utility for each pool set point as follows:

$$x^*_{LC} = \arg\max 1 - P(\hat{y}|x) \tag{2.16}$$

where $\hat{y}$ is the most probable label for x under the current model. By subtracting $P(\hat{y}|x)$, we are compute the probability of the least probable label for each pool set point. From this, the learner can then query points for which the model has the highest least confidence.

Random Acquisition

This acquisition function is typically considered as a baseline comparison for all proposed active learning algorithms. Most previous research on active learning shows that the proposed algorithm can outperform the random acquisition function. While previous research considered classifiers other than CNNs, in this framework, we implement the random acquisition function based on CNNs. At every acquisition iteration, points are randomly added for training the CNN model. We evaluate this acquisition function, and compare whether our proposed acquisition functions can perform better achieving a higher level of accuracy with few labelled samples.

In the next section, we discuss few related work which can also be used to represent uncertainty in a deep learning framework. However, unlike the methods discussed below, the

dropout uncertainty tool from [9] is the only easily extendable framework for extending to CNNs. For our work in this thesis, we therefore use the dropout uncertainty as approximate Bayesian inference for obtaining uncertainty estimates required for active learning. We include a discussion of other related approaches in the next section.

2.4 Related Work

Previously we mentioned the importance of obtaining good estimation of uncertainty for our dropout acquisition functions. We discussed how our proposed acquisition functions uses test-time dropout for obtaining estimates of uncertainty over images using a Bayesian CNN framework. In section 2.4.1 we discuss related research for obtaining uncertainty estimates and avoiding overfitting in deep learning using a Bayesian Neural Network framework. However, compared to our approach, these methods have not yet been shown to work well on CNNs when considering high dimensional inputs such as images. Most of the related approaches considered below, even though shows that these models can give good predictive output distribution, however their extensions to CNN models have not be done yet. We re-emphasize the ease with which test-time MC dropout can be applied to a Bayesian CNN model to obtain good uncertainty estimates for active learning. This is important since in our considered framework, computation time is of importance, as we are dealing with repeated training of a deep model. The MC dropout approach of [9] can give model uncertainty without increasing model complexity or the number of parameters, which plays a significant role in the active learning setting for deep learning.

In chapter 3, we will demonstrate the reliability of our dropout uncertainty estimates compared to some of the related work mentioned below. In particular, we will compare several frameworks that can represent uncertainty efficiently using an active learning regression task where pool points with highest variance are queried. The results in chapter 3 will show that while uncertainty estimates can be obtained for several methods used here, they can only be used in the regression active learning task, with constrains on input dimensions. Unlike other methods, the dropout uncertainty fraemwork proposed by [6, 9] is the only easy to implement approach that can be extended for CNN models for dealing with image classification tasks.

2.4.1 Approximate Bayesian NNs and DGPs for Uncertainty Estimates

Bayesian Neural Networks and Variational Inference

It has been known that a neural network with infinitely wide hidden units with distributions placed over their weights corresponds to the Gaussian Process model [5]. Furthermore, models such as Bayesian Neural Networks have been studied extensively with finite NNs having distributions placed over their weights [5], [4]. These models can offer robustness to over-fitting and uncertainty estimates for neural networks, but there are severe computational costs and challenging inference to it. Variational inference has been proposed for neural networks, but without much success [19] largely due to the difficulty of deriving analytical solutions to the required integrals over the variational posteriors. Such solutions have been shown to be complicated for even the simplest of the network architectures such as single layer feedforward networks with linear outputs [19], [20]. A recent approach applied variational inference to neural networks [21] which introduces a stochastic variational method that can be applied to most neural networks. There has been recent advances in these methods introducing sampling-based variational inference and stochastic variation inference [22], [23], [24]. In [24], the ideas of deep neural networks and approximate Bayesian inference were combined for deriving directed generative models for scalable inference and learning. Furthermore, there has been approaches to obtain new approximations for Bayesian Neural Networks which have been shown to perform as well as dropout [25]. In [25], a backpropagation compatible algorithm was introduced called Bayes by Backprop for learning probability distributions on the weights of the neural network. It introduces a new algorithm for learning neural networks with uncertainty on the weights and shows that the algorithm is comparable to that of dropout. By introducing a principled algorithm for regularisation built upon Bayesian inference on the weights of the network, [25] demonstrates that this uncertainty can improve predictive performance on regression problems by expressing uncertainty in regions of fewer or no data. However, these models have high computational cost for obtaining uncertainty estimates. In orer to represent uncertainty in these models, the number of parameters in these models is doubled for the same network architecture, while also requiring more time to converge. Therefore, these models introduces additional computation which are further expensive, in order to obtain uncertainty estimates. Furthermore, [25] demonstrates uncertainty estimates over regression problems using neural networks while in our work, we consider uncertainty estimates over image data using Bayesian CNNs. All the approaches above have been shown to work on a Bayesian Neural Network implementation, and little work has been done to extend these algorithms for CNN models.

Expectation Propagation and Probabilistic Backpropagation

An alternative approach to variational inference is to consider the use of expectation propagation [26] which have been shown to improve on the uncertainty estimates compared to VI approaches. Deep neural networks trained with backpropagation typically have the disadvantages such as the need to tune a large number of hyperparameters, tendency to overfit the training data, and models with backpropagation do not give a calibrated probabilistic predicition. Furthermore, Bayesian techniques discussed aboe lack the ability to scale to large datasets and network architectures. [26] therefore introduces a scalable method for learning Bayesian neural networks called Probabilistic Backpropagation (PBP) and shows that PBP provides accurate estimates of the posterior variance on the network weights. Bayesian approaches to neural networks can automatically infer the hyperparameter values by marginalizing them out of the posterior distribution, and can also naturally account for uncertainty in the parameter estimates and can propagate this uncertainty into predictions. [26] offers a probabilistic appraoch to backpropagation algorithm by propagating probabilities forward through the network to obtain marginal likelihood and then propgating the gradients of the marginal likelihood backwords. By using this probabilistic approach to backprop, PBP can produce calibrated uncertainty estimates of the posterior uncertainty in the network weights, and also offers robust overfitting since they average over parameter values instead of choosing a single point estimate. [9] compares the dropout approach to obtaining uncertainty estimates with PBP and shows a significant improvement in RMSE and uncertainty estimation. While the approach taken by PBP is comparable to our work, and have been shown to work on both classification and regression problems, such Bayesian approaches to neural networks have not been shown to work well considering high dimensional inputs such as images. PBP works only on low dimensional classification settings, and have shown results for active learning classifiers. However, PBP have not yet been shown to work well on CNNs to obtain uncertainty estimates when considering image data for active learning.

Deep Gaussian Processes

Deep Gaussian Processes (DGPs) are multi-layer hierarchical generalisations of Gaussian Processes and are equivalent to neural networks with multiple infinitely wide hidden layers. [27] develops an approximate Bayesian learning scheme to enable DGPs to be applied on large scale regression problems using an approximate Expectation Propagation scheme. Their approach further uses the probabilistic backpropagation algorithm for learning to show that such methods are better than sampling-based approximate inference methods for Bayesian

neural networks. By using DGPs, [27] shows that these nonparametric probabilistic models offers a greater capacity to generalise and can provide better calibrated uncertainty estimates than alternative deep models. [27] focuses of Bayesian learning of DGPs which involves inferring the posterior over the layer mappings and hyperparameter optimisation via the marginal likelihood. However, results on DGPs only shows initial work on classification, but does not show significant gain over GP. Additionally, DGPs or GPs have not yet been shown to work well on high dimensional inputs and it is computationally much more expensive to train these models for image data to get uncertainty estimates. However, our approach to using Bayesian CNNs can be very easily used to obtain uncertainty estimates over images for an active learning setting by only applying dropout at test time. There are significant disadvantages to using DGPs, especially considering the approximate EP framework, and the difficulty of training DGPs on high dimensional inputs.

2.4.2 Other Acquisition Functions for Images

Several methods have previously been proposed for active learning algorithms for images, since for images and videos providing training data is expensive in terms of human time and effort. However, most of these approaches are based on commonly used machine learning models such as SVMs. No previous work for active learning of images had been used considering CNN models due to CNNs being prone to overfitting with small datasets. [28] previously proposed acquisition functions based on uncertainty sampling where they used an uncertainty measure that generalises margin based uncertainty and used a SVM classifier for multi-class classification. Similarly, [29] developed entropy based active learning where the learner chooses an image to label that maximizes the expected amount of information again about the set of unlabeled images. Their approach called "Minimum Expected Entropy", although used an entropy based active learning framework to measure informativeness, used a committee of K-NN and SVM classifier to estimate class probabilities for the unlabelled images. Unlike their approach, we use the deep learning framework for the use of Bayesian CNN models, since CNNs have been shown to achieve state of the art performance for images [1]. Furthermore, [30] combined the information density and most uncertainty measure together to select query points for image classification. To the best of our knowledge, no previous method had therefore been used using CNN models. In this work, we therefore demonstrate the effectiveness of Bayesian CNNs for active learning in image classification tasks.

2.5 Combining Active and Semi-Supervised Learning

In this section, we take a different approach to our work. We consider the idea of combining active learning and semi-supervised learning, extending work from [31] by using CNN models which was previously not considered. Further from [31], we combine the two fields under a Gaussian random field model, but instead using a CNN model architecture for a classifier. We begin by describing the combined active learning and semi-supervised learning framework of [31] formulated with a graph-based semi-supervised learning approach and a Gaussian random field.

In the semi-supervised learning approach, we again use labelled and unlabelled datasets L and U, and construct a graph $G = (V, E)$ where the nodes correspond to the n data points. The edges are represented by a *nxn* weight matrix W which is given by a radial basis function (RBF) with weights $w_{i,j}$. We consider nearby image points in the Euclidean space. While [31] considered a relaxation of the requirement that labels should be binary, we experiment with both binary and multi-class labels. The approach of [31] is based on using harmonic energy minimizing functions where a low energy corresponds to a slowly varying energy function over the graph. Since we want unlabelled points that are nearby in the graph to have similar labels, the energy function is defined as:

$$E(y) = \frac{1}{2}\sum_{i,j} w_{i,j}(y(i) - y(j))^2 \tag{2.17}$$

The minimum energy function is therefore given as $f = argmin_{y|L=y_L} E(y)$ and this harmonic energy minimizing function can be computed in terms of matrix methods. Defining the diagonal matrix $D = diag(di)$ where $d_i = \sum_j w_{ij}$ and the combinatorial laplacian is the *nxn* matrix given by $\Delta = D - W$, then the laplacian matrix can be partitioned into blocks given by:

$$\Delta = \begin{bmatrix} \Delta_{ll} & \Delta_{lu} \\ \Delta_{ul} & \Delta_{uu} \end{bmatrix} \tag{2.18}$$

and if we let $f = \begin{bmatrix} f_l \\ f_u \end{bmatrix}$ then the solutiona of the mean harmonic energy function for the unlabelled points is given by

$$f_u = -\Delta_{uu}^{-1}\Delta_{ul} f_l \tag{2.19}$$

By forumulating the semi-supervised learning problem in terms of a Gaussian random field on this graph, we can then perform active learning on top of this similar to as defined by [31]. Similar to [31], we propose to perform active learning with the Gaussian random field model by greedy querying points so as to minimize the risk of the harmonic energy minimization function. We also consider the risk to be the estiamted generalisation error of a Bayes classifier. More on this semi-supervised learning framework can be found in [31]. However, in constrast to the approach taken by [31], while we similarly query points which minimizes the risk, after querying points from the pool set, we evaluate the final output using a CNN classifier with a softmax output.

The active learning approach based on minimizing the risk of the harmonic energy function on graph-based semi-supervised learning is defined as follows. Similar to [31], we compute the estimated risk as $\hat{R}(f) = \sum_{i=1}^{n} min(f_i, 1 - f_i)$. If we perform active learning and query a point x_k, y_k, then this point will also change the Gaussian field and its mean energy function. Denoting the new harmonic function to be $f^{(x_k, y_k)}$, then the changed estimated risk will be given by

$$\hat{R}(f^{+(x_k,y_k)}) = \sum_{i=1}^{n} min(f_i^{(x_k,y_k)}, 1 - f_i^{(x_k,y_k)}) \tag{2.20}$$

but since we do not know y_k for the pool point before it is queried, we assume the estimated risk to be approximated by

$$\hat{R}(f^{+x_k}) = (1 - f_k)\hat{R}(f^{+(x_k,0)}) + f_k\hat{R}(f^{+(x_k,1)}) \tag{2.21}$$

and the active learning criterion for a binary classification task as defined by [31] is to choose the next query that minimizes the estimated expected risk

$$k = \arg\min_{k'} \hat{R}(f^{+x_{k'}}) \tag{2.22}$$

We extend the work from [31] to a multi-class classification setting for image classification task, by similarly combining the active and semi-supervised learning framework. This extension can be easily made by defining the expected estimated risk to be simply

$$\hat{R}(f^{+x_k}) = f_k\hat{R}(f^{+(x_k,y)}) \tag{2.23}$$

and similarly query the next point which minimises the energy function following equation 2.22. However, the only difference in our work is that we evaluate the output of the active learning algorithm using a traditional CNN classifier with a softmax output. As defined

above, we similarly compute the harmonic energy function and the estimated risk for both the binary and multi-class setting, but instead evaluate the output with a CNN classifier.

In the experimental results section, we will evaluate the performance of this Gaussian random field harmonic energy based active learning criterion on image classification task. More importantly, we will compare our dropout uncertainty acquisition functions with this combination framework to evaluate which method performs better. The framework described in this approach, based on extension from [31] is a more computationally expensive task compared to our dropout active learning approach, since this involves computing the estimated risk for all the points in the pool set. We will evaluate this scheme in the experimental results section, first for a binary classification task, and then extended for multi-class classification.

Chapter 3

Experimental Results and Analysis

In this chapter, we demonstrate our experimental results and present the effectiveness of our proposed Bayesian active learning acquisition functions based on using the Bayesian CNN architecture. We illustrate that by using model uncertainty casting dropout training in neural networks, we can perform information theoretic Bayesian active learning with Bayesian CNNs. We show that a significant improvement in classification performance can be achieved even with training Bayesian CNN models with very few labelled training data. Since our framework is the first to propose active learning methods using CNNs for images, we present state-of-the-art predictive performance in using active learning in the deep learning framework.

We illustrate the importance of obtaining good model uncertainty estimate by comparing the dropout acquisition functions with softmax based methods which do not capture model uncertainty. We inspect the use of different model architectures and non-linearities in the Bayesian CNN model which corresponds to different GP covariance functions to capture uncertainty. Our results on MNIST demonstrates the importance of model architectures and non-linearities, which affects the performance of the active learner quite significantly. We further demonstrate the reliance of our dropout uncertainty estimate for active learning by comparing with several methods (such as other approximate Bayesian NNs and DGPs) on a simple active learning regression task. We also compare our proposed algorithms with approaches that combines active learning with graph-based semi-supervised learning for images on a binary image classification tasks. Finally, we include a summary of our experimental results and illustrate that our active learning approach in the deep learning framework achives state-of-the-art performance.

3.1 Experimental Setup

We show the performance of our dropout Bayesian CNN based acquisition functions on the MNIST dataset. We perform dropout after all convolution and weight layers in the LeNet5 CNN model architecture to capture model uncertainty. All our experimental results are averaged over 5 experiments. In the active learning experimental setup, we initially start with only 20 training data points and fit a model on this dataset. We ensure that the initial training set of 20 datapoints consists of a uniform distribution of all classes to ensure that the initial model is trained with all classes of images. We validate on 10,000 labelled samples, and our setup has a pool set of 40,000 points from which to select our query points to be added to the training set. Further to using dropout during training and test time, we further add a L2 regulariser in the top NN layer of the CNN architecture, with a weight decay parameter to be fine-tuned by cross validation. Our model uses the ADAM optimizer [32], and we use 50 training epochs for every training label set with a batch size of 128. Unless otherwise state, we use the ReLU activation function for the non-linearity in the Bayesian CNN models. At every acquisition iteration, we subsample 2000 points from the pool set for which to estimate the predictive distribution from MC dropout samples, and we use this pool subsample to query the point to be added to training set. Every time a point x is selected, we delete this pool point from the pool set and add it to the training set. The CNN model architecture is re-trained after every pool point acquisition and the test set accuracy is evaluated using 10,000 test samples. Unless stated, we follow the same experimental setup discussed above, and evaluate the performance of our active learners on the MNIST dataset using 10,000 test samples. All our experiments were done using the Keras framework [33].

The experiment configuration files, scripts and results are available at https://github.com/Riashat/Active-Learning-Bayesian-Convolutional-Neural-Networks.

3.2 Performance of Acquisition Functions

3.2.1 Experimental Results

In this section, we evaluate the performance of each of our dropout based acquisition functions on the MNIST dataset. In section 3.2.1 below, we show the performance of each of our active learner on the 10,000 MNIST test samples, starting with 100 training datapoints. The focus of the experiments below is to demonstrate that the Bayesian CNN models can avoid overfitting on the small dataset. For every query point added to the training set, we show the training and validation accuracy plots to ensure that overfitting is avoided for each active learning

acquisition from the pool set. We present the experimental results for each of our dropout acquisition functions using the Bayesian CNN implementation. Note that it is important to analyse model fitting issues for every active learning acquisition iteration. Since we are dealing with small training datasets for our Bayesian CNN models, we need to illustrate that these models casting dropout as approximate Bayesian inference can avoid model overfitting.

Dropout Bald

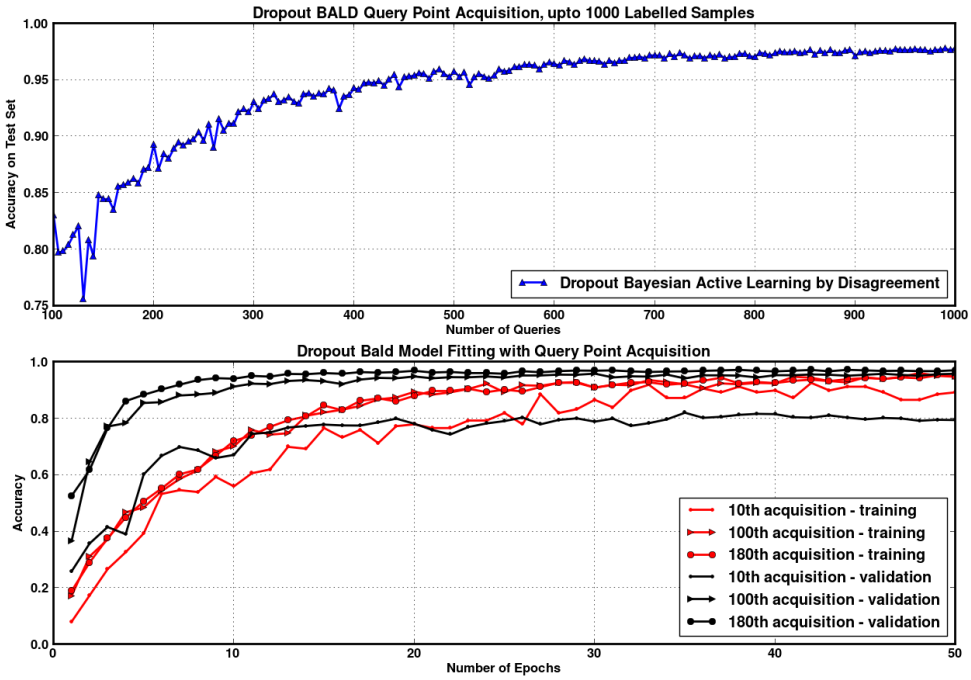


Fig. 3.1 Performance of the active learning algorithm using Dropout BALD acquisition function on MNIST. Model Fitting on small training dataset using Bayesian CNN framework

Figure 3.1 shows how the performance of the Bayesian CNN classifer improves with the number of queries made by the active learner. The subplot further shows that the CNN models avoid overfitting even when trained with a very small dataset. By using the uncertainty information from MC dropout samples, the Dropout BALD acquisition function generalises quite well on the unseen data. The model fitting results in figure 3.1 are shown only for few acquisitions, notably the acquisitions at the beginning and towards the end. The model achieves a better fit at the $180th$ acquisition iteration compared to the $10th$ acquisition

iteration. Most importantly, the Bayesian CNN model does not overfit at any of the active learning acquisitions as illustrated by figure 3.1 below.

Dropout Variation Ratio

Figure 3.2 shows the performance of our Dropout Variation Ratio active learning algorithm, illustrating the significance of robustness to model fitting in small data regime. Figure 3.2 shows that even though the model is slightly prone to overfitting for the $10th$ acquisition iteration, where we only have 200 training samples, it becomes less prone to overfitting for the $180th$ acquisition iteration. However, it is important to note that even for 200 training samples, the model does not overfit. As illustrated in [6], this is the benefit of using Bayesian CNN compared to a traditional CNN, as the Bayesian approach makes the model robust to overfititng issues.

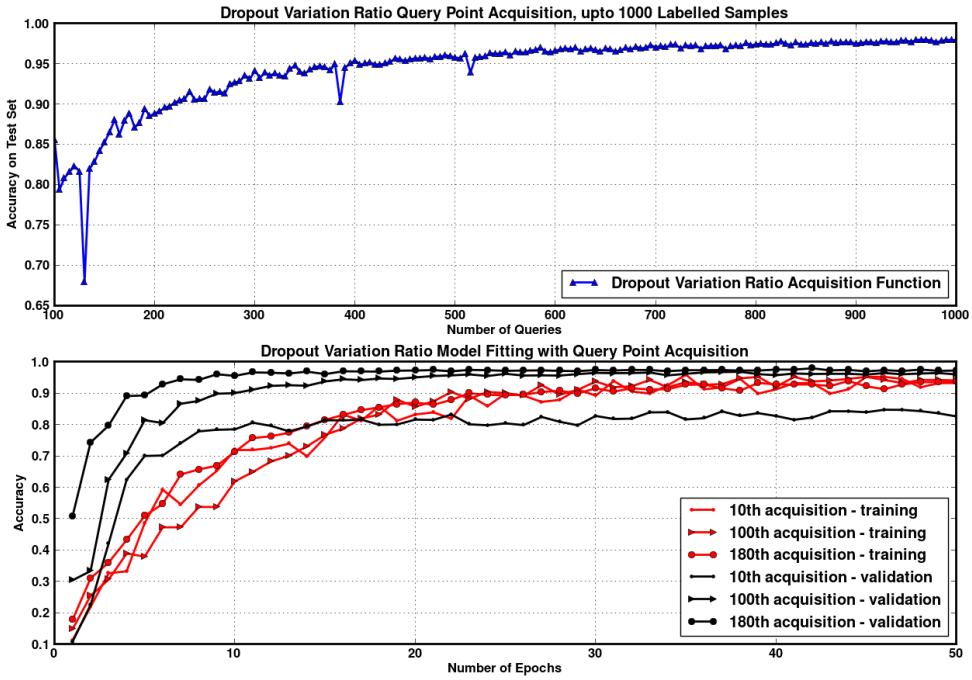


Fig. 3.2 Test accuracy and model fitting using Dropout Variation Ratio acquisition function

Dropout Maximum Entropy

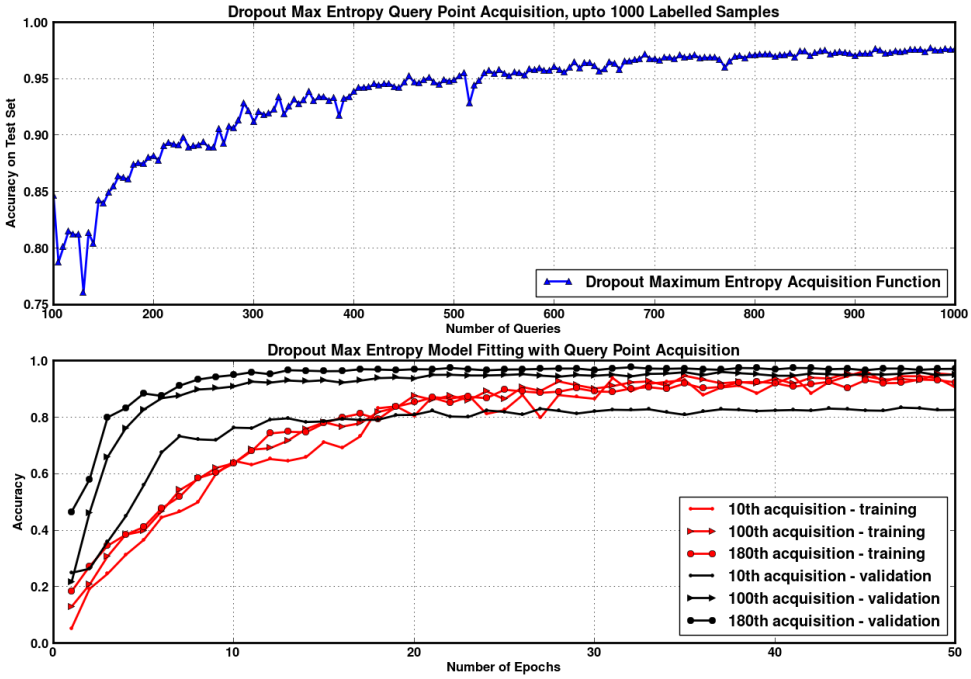


Fig. 3.3 Test accuracy and model fitting using Dropout Max Entropy acquisition function

We then implement our Dropout Maximum Entropy acquisition function. Similar to the commonly used approach based on querying points with maximum entropy, the only difference with our approach is that we use the mean of the predictive distribution to compute the entropy, instead of simply taking the predicted probabilities. In later section, we will furhter demonstrate how our Dropout Max Entropy acquisition function can outperform the baseline maximum entropy based acquisition functions, since our approach using Bayesian CNNs can make the model less prone to overfitting as illustrated below.

Dropout Bayes Segnet

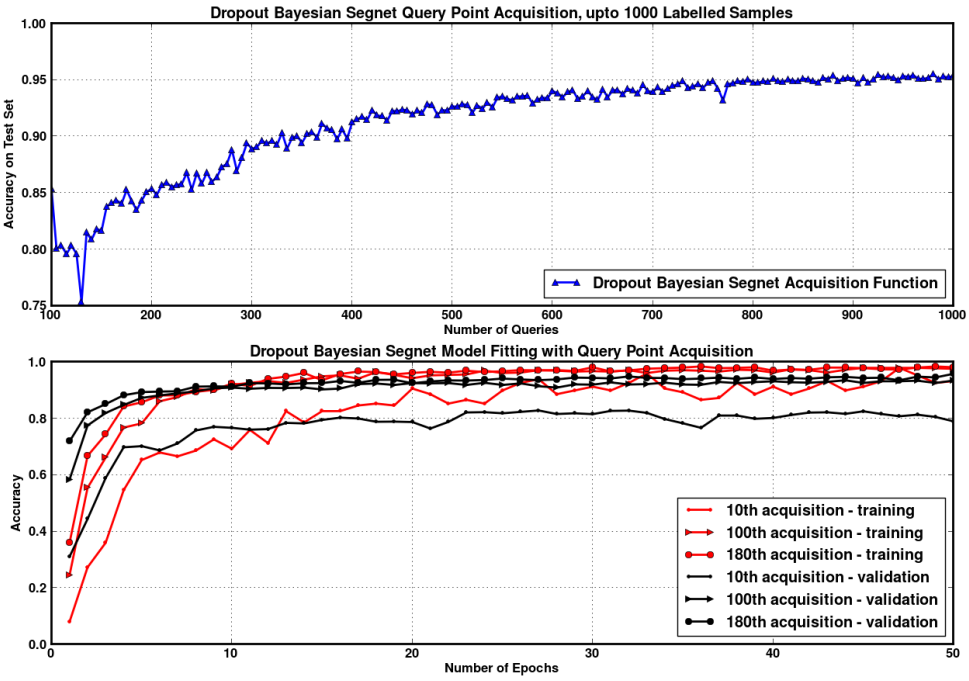


Fig. 3.4 Test accuracy and model fitting using Dropout Bayes Segnet acquisition function

3.2.2 Discussion

The experimental results in this section illustrates that our active learning algorithms avoids overfitting for each acquisition iteration using the Bayesian CNN model. We illustrate the performance of each of our proposed acquisition functions on the MNIST dataset, and demonstrate that unlike traditional CNN models, the Bayesian approximation to dropout can significantly avoid overfitting for training CNN models with small dataset. For each of the acquisition functions, we show the performance on the test set, along with the validation plots to illustrate model fitting.

In the next section, we will compare our proposed active learning algorithms with baseline acquisition functions typically used in active learning. For the baseline functions, we use a traditional CNN model architecture, and compare our methods based on using Bayesian CNNs.

3.3 Comparison of Acquisition Functions

We then compare our proposed acquisition functions with other acquisition functions typically used in active learning. In particular, we compare our proposed dropout Bayesian CNN active learning algorithms with other baseline acquisition functions used (random, maximum entropy and maximum margin). As stated previously, we start with 20 training data points and query upto 1000 points. This means, our model is trained with a final labelled set of 1000 training samples, and tested on 10,000 samples. Note that, instead of querying only 1 point at a time from the pool set, here we query 10 points at each iteration. This is also to avoid too many repeated training of CNN models which requires computational resources and time.

In a later section, we will demonstrate the significance of querying 1 points or higher number of points at time from the pool set. We also compare our MC dropout functions with softmax functions typically used in CNN models. [9] further discusses the significance of softmax output compared to passing a distribution through a softmax. In our results below, we further justify the importance of uncertainty estimate in active learning by comparing MC dropout with standard softmax outputs. [9] shows that the predictive probabilities obtained from the softmax output cannot be interpreted as model confidence since a model can be highly uncertain about its predictions even with a high softmax output.

The experimental results in this section illustrate that our proposed acquisition functions for active learning can significantly outperform the other baseline functions on the MNIST image dataset. However, by comparing our proposed functions by itself, we note the importance of using good uncertainty estimates for active learning. As illustrated later, we see that our Dropout BALD and Dropout Variation Ratio acquisition functions can outperfrom Dropout Bayes Segnet and Dropout Maximum Entropy. This is mainly because taking the maximum entropy as a measure of most uncertain point is perhaps not a good measure since the entropy values are also affected by the probability distribution of all the classes. Furthermore, as discussed earlier, our Dropout Bayes Segnet function uses standard deviation of probabilities as an uncertainty measure, which is not a good measure. The experimental results below demonstrates this.

3.3.1 Experimental Results

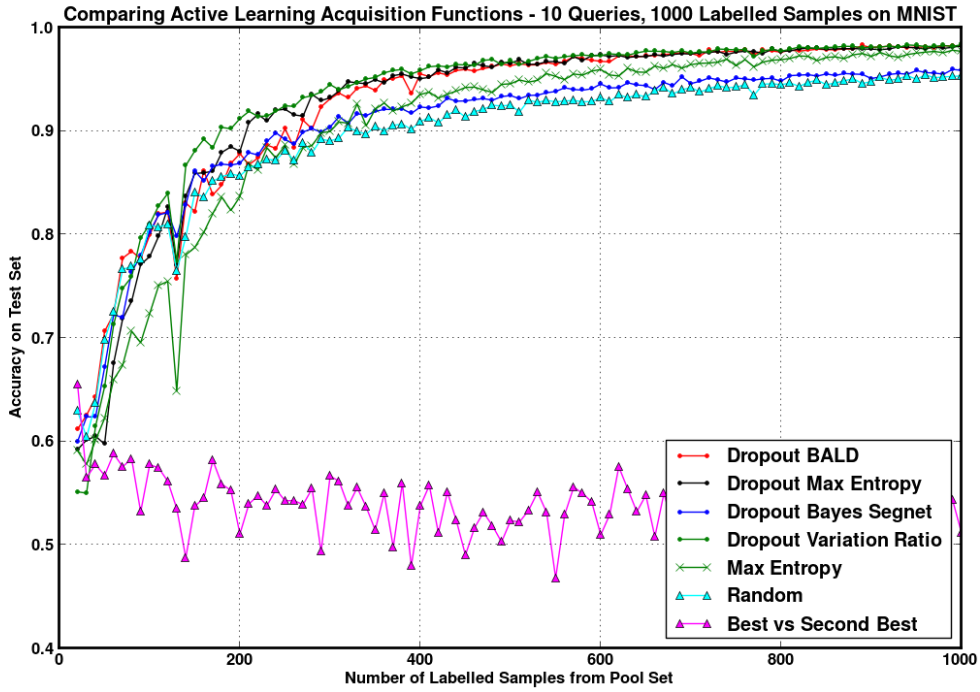


Fig. 3.5 Comparison of MC dropout acquisition functions with Baseline acquisition functions

At first, we simply compare our proposed algorithms with baseline functions. Figure 3.5 compares the MC dropout uncertainty estimate based Bayesian CNN acquisition functions with other baseline functions commonly used in active learning. Result in figure 3.5 demonstrates the usefulness of using our proposed active learning acquisition functions. However, result in figure 3.5 does not necessarily show whether model uncertainty is required for active learning, since it maybe that our method outperforms simply due the effectiveness and properties of the acquistion function such as BALD. However, in figure 3.6 we further illustrate that this is otherwise. Figure 3.6 the significance of using MC dropout uncertainty estimates. We show that the MC dropout based acquisition functions can outperform the softmax based functions, which simply uses the class predictive probabilities from a single stochastic pass. We will demonstrate the importance of uncertainty estimates in more details in a later section.

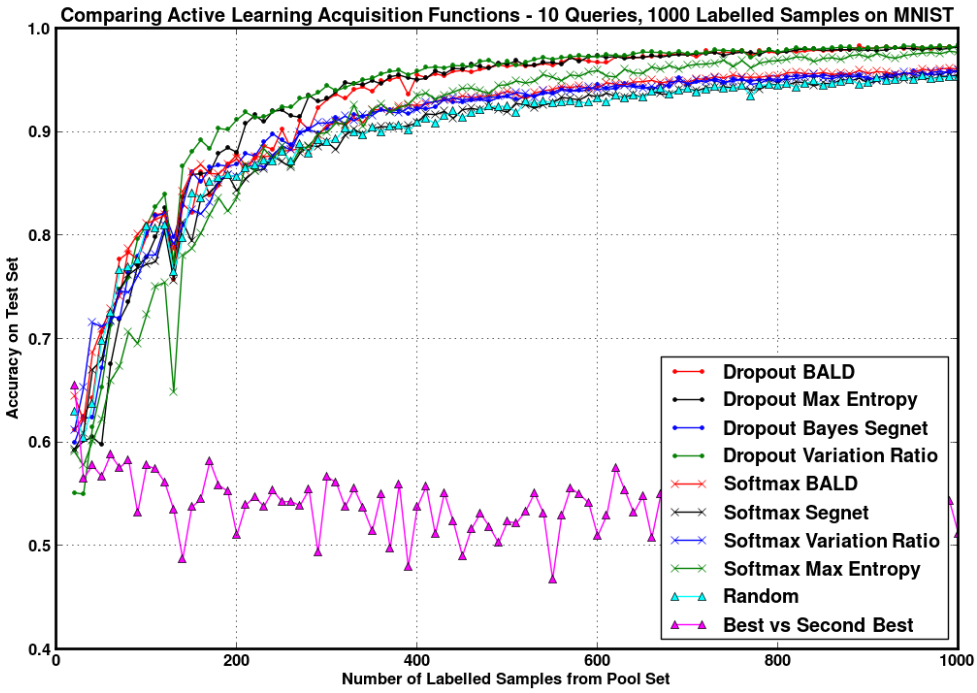


Fig. 3.6 Significance of uncertainty estimates : Comparison of acquisition functions using MC dropout samples and softmax output

Figure 3.6 further shows the comparison of our active learning algorithms with a traditional CNN architecture with a softmax output. For example, in Softmax BALD, the same acquisition or utility function is used similar to BALD, with the difference that Dropout BALD uses MC samples to obtain a predictive distribution through a softmax output, whereas Softmax BALD the predictive probability obtained from a softmax output of a CNN architecture. This result is further illustrated in the next section.

Querying even fewer datapoints - Upto 100 samples

In order to achieve data efficiency, we further looked into the significance of querying even few points (upto 100 instead of 1000) and demosntrate how our model performs when trained with even fewer labelled samples. Figure 3.7 below further demonstrates how the test set accuracy on MNIST depends when the model is trained with even fewer labelled samples. Note that the result in figure 3.7 maybe affected by model overfitting issues since we have too few training data to train the Bayesian CNN models. For future work, one interesting direction would be further achieve a high predictive performance even if the model is trained with upto 100 training labelled samples only.

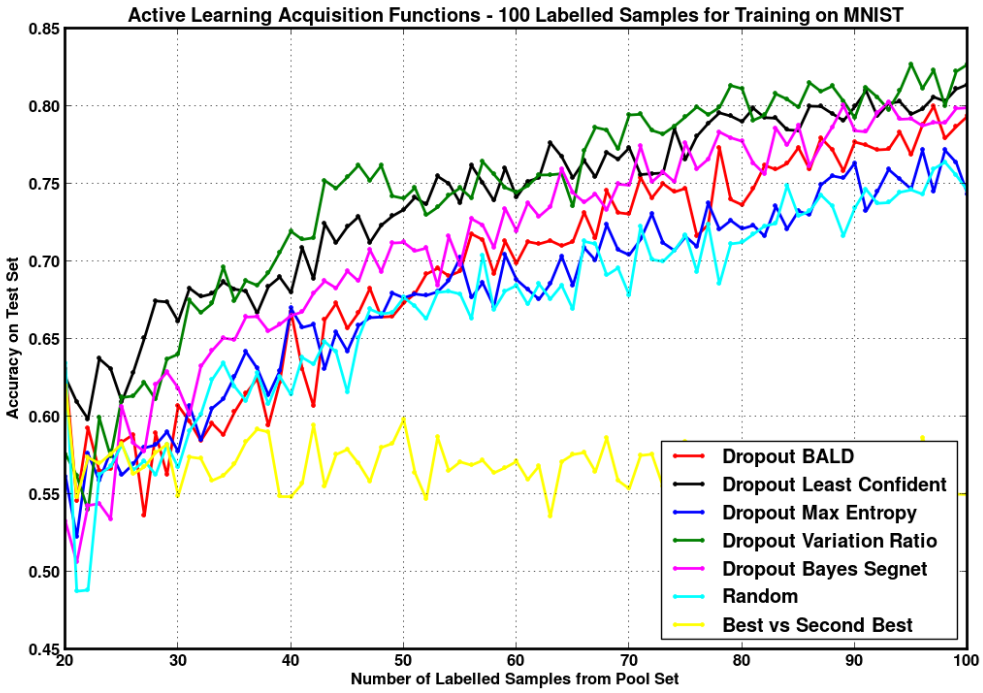


Fig. 3.7 Querying upto 100 labelled samples and validating on 10,000 samples on MNIST. Significance of using fewer labelled samples for training

Significance of Weighted Inputs in small data active learning

We further demonstrate the significance of using weighted inputs in the Bayesian CNN. If we are starting with only 20 training points, and querying upto 80 points for a total of 100 samples only for training the Bayesian CNN model, it is too small a dataset for training, to further validate on 10,000 samples. Furthermore, if we are querying only one point at a time and adding to the CNN model for training, these points are often smoothed out by the network, leading to no significant overall change.

One way to avoid this is to weight the inputs in the objective function. In other words, this means that we make the model train more significantly using the recently added points, and weigh out the previously added points in a decaying manner. This makes the model place a higher importance to training with the most recent data points. This is particularly useful in active learning since our most recent points are often the most informative points to train on. Figure 3.8 below demonstrates the significance of weighing the inputs with different γ parameters, where γ defines the weighing proportion. A higher γ means that previously added points are placed a higher importance for training, whereas lower γ means we put

very less importance to the previously added points to the training set in the active learning acquisition iterations.

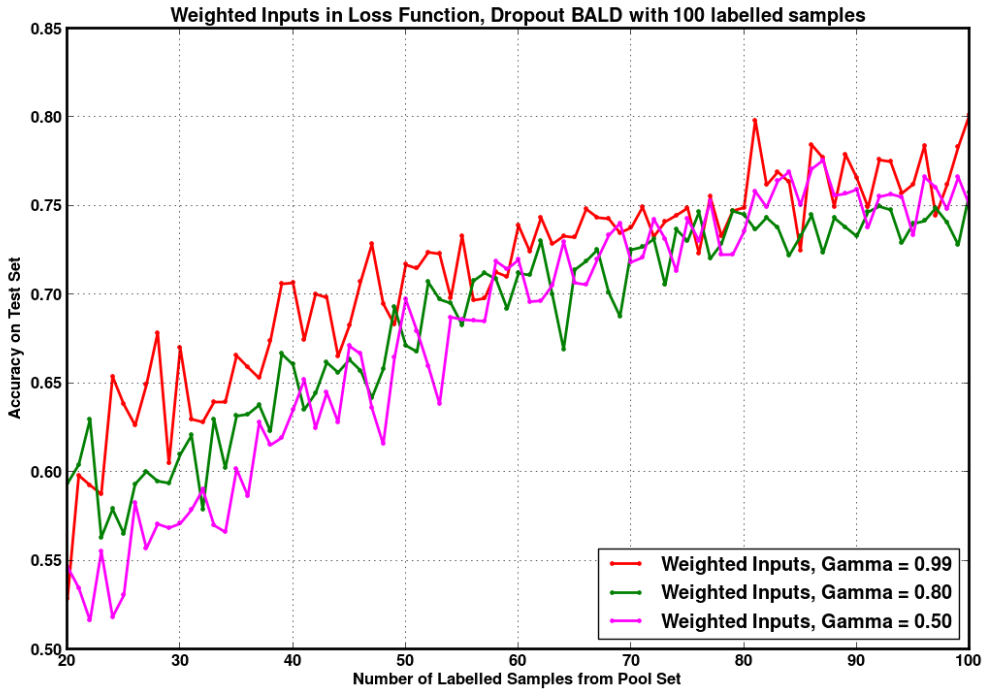


Fig. 3.8 Significance of using weighted inputs in the loss function for training Bayesian CNN with very small training dataset

Figure 3.8 illustrates that a very low value of γ is not useful. A higher γ value of 0.99 dominates, since this means the weights added to previous points are not too small compared to the recently added points. However, note that the results in figure 3.8 are based on training the model with very small data and validating on $10,000$ test samples, which often makes the model prone to overfitting.

3.3.2 Discussion

The experimental results in this section illustrates the significance of our proposed acquisition functions, compared to other baseline functions typically used. Figure 3.5 shows that the MC dropout acquisition functions can significantly outperfrom the maximum entropy and random acquisitions. Further to this, figure 3.6 shows that even when applying the same acquisition function, the uncertainty estimates from MC dropout samples to obtain a predictive distribution plays an important role. Due to a much better uncertainty estimate

obtained from MC dropout, these acquisition functions typically outperform the softmax outputs of a traditional CNN architecture. This further demonstrates the significance of usina a Bayesian CNN implementation compared to a traditional CNN for active learning. Here, also note that our Dropout Bayes Segnet performs as poorly as random acquisition. This is also because, as discussed previously, standard deviations of probabilities is not a good measure of uncertainty. This is further justified from the results in this section. Since Dropout BALD can significantly outperform Dropout Bayes Segnet, it further demonstates the importance of good uncertainty estimates for use in active learning. Finally, figure 3.7 shows the significance of querying even fewer data points from the pool set. Figure 3.7 shows that even though the test set accuracy improves with every informative query point added to the training set, it does not necessarily achieve same test accuracy. This is also because 100 training points for a CNN model might be too less (compared to using 1000 points) for measuring their test performance on 10,000 samples.

3.4 Representing Model Uncertainty in Deep Learning for Active Learning

In section 3.3 we demonstrated the performance of our MC dropout active learners compared to other acquisition functions. We demonstrated that an active learner based on Bayesian CNN implementation can outperform a traditional CNN based active learner, even when using the same BALD acquisition function. In this section, we further demonstrate this in details.

In particular, we compare the estimates obtained with and without using dropout, and following our the same criterion for our proposed acquisition functions. We evaluate all our proposed acquisition functions with and without using test-time dropout, and evaluate the performance of these models on MNIST test data again to further justify the importance of the uncertainty estimates for active learning. Here, we want to demonstrate the significance of model uncertainty in active learning, which can be obtained from a Bayesian CNN based active learning algorithm compared to a traditional CNN architecture.

3.4.1 Experimental Results

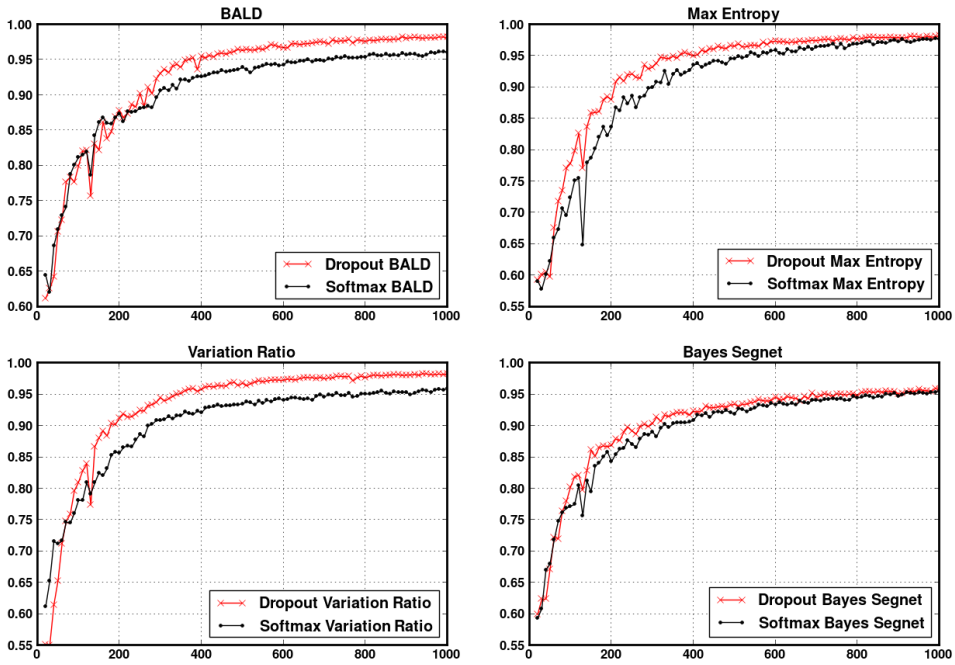


Fig. 3.9 Comparison of active learning with Bayesian CNN vs traditional CNN (with and without using test-time MC dropout samples)

Figure 3.9 compares our proposed acquisition functions for a Bayesian CNN implementation compared to a traditional CNN output. In other words, the Dropout acquisition functions are based on achieving model uncertainty from a Bayesian CNN, whereas the Softmax functions simply use output a traditional CNN. Our experimental results in figure 3.9 shows that the dropout uncertainty based acquisition functions (shown in red) can outperform the softmax based functions for all four of our proposed algorithms. This further validates the importance of using MC dropout samples to obtain a predictive distribution, since the model uncertainty obtained from approximate Bayesian inference in CNNs can not only avoid over-fitting for small datasets, but can also significantly improve the overall predictive performance of our active learners. Furthermore, note how the Dropout Bayes Segnet and Softmax Bayes Segnet performs almost equally. This again demonstrates that the Bayes Segnet approach does not give us good uncertainty estimates for use in active learning. In contrast, having a good estiamte for BALD and variation ratio based acquisition functions is of importance in active learning.

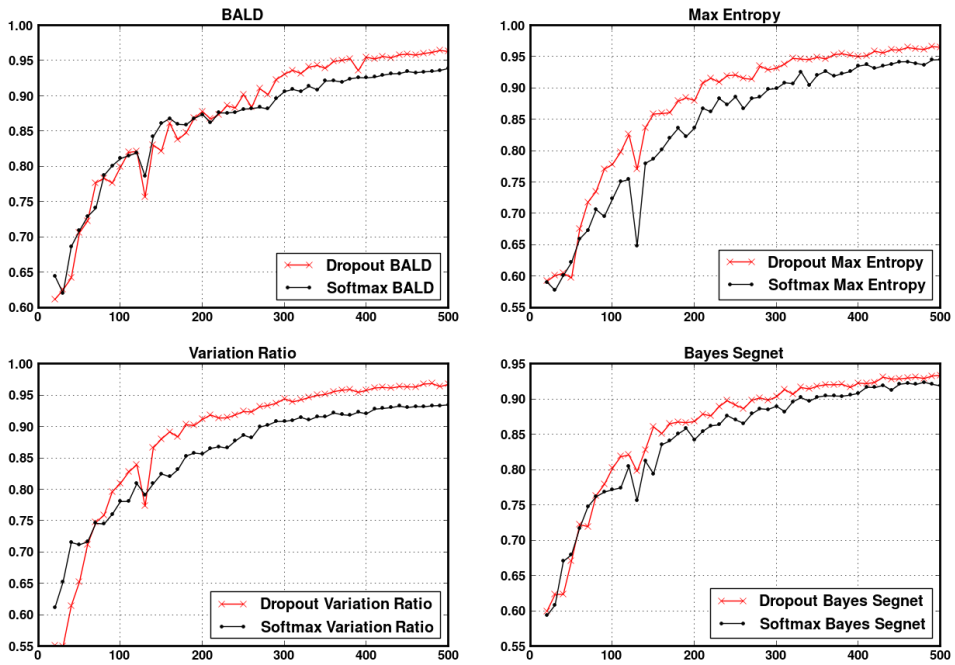


Fig. 3.10 Demonstrating the importance of good uncertainty estimates in small data settings for active learning

Figure 3.10 below further demonstrates the results above in small data settings. The comparison between the active learning algorithms based on with and without using test-time dropout can be seen more significantly in the small data setting, querying only upto 500 points for training instead of 1000. When querying only upto 500 labelled training samples, it is far more clear of how the dropout acquisition functions can outperform the softmax ones. This further justifies that using a softmax at the output layer of a CNN does not give us model uncertainty unlike using test-time dropout.

3.4.2 Discussion

The experimental results in section 3.4 above demonstrates the importance of a good uncertainty estimate for use in active learning. Figure 3.9 shows that the MC dropout model uncertainty estimates in Bayesian CNN plays a significant role for improving the performance of our active learner, compared to using a traditional CNN model. Note how the differences are more signifincant for the BALD and Variation Ratio based acquisition functions, compared to Maximum Entropy and Bayes Segnet. The results here also draws an important comparison between the performance of each of our acquisition functions as well. From here,

we can justify that BALD and Variation Ratio are better utility functions compared to simply taking the maximum entropy point from the pool set. It also further demonstrates that the standard deviations of probabilites is not a good measure of uncertainty, which is justified from the maximum test accuracy reached by each of the active learners. The Bayes Segnet based acquisition function performs poorly compared to Dropout BALD and Variation Ratio.

3.5 Bayesian CNN Model Architectures and Non-Linearities for Active Learning

In this section, we further demonstrate the significance of different Bayesian CNN model architectures and non-linearities for use in active learning. [9] suggested that the combination of NN non-linearities and weight regularisation would correspond to different GP covariance functions for uncertainty estimates. For example, L2 regularisaiton might be more appropriate if we want the uncertainty to increase away from the data. In this section, we further demonstrate how the use of different CNN model configurations and activation functions can change the predictive mean and variance obtained from the output of the Bayesian CNN model. We investigate the change in uncertainty calibration for diffrent configurations, for choosing the best architecture that can give a reliable uncertainty estimate for use in active learning. For our Dropout BALD acquisition function, we used different non-linearity at every layer of the Bayesian CNN model architecture. Our results in this section demonstrates the importance of choosing the right model architecture and non-linearity for use in active learning. This is in similar line as to how choosing the covariance function for GPs plays an important role in the uncertainty estimates that GPs have to offer.

3.5.1 Experimental Results

We use only the Dropout BALD active learning algorithm for demonstration of the significance of model architectures. Here, we start with 100 training points, query 10 points at each iteration, query upto 1000 points and evaluate the performance on 10,000 MNIST test samples.

3.5.2 Bayesian CNN Non-Linearities

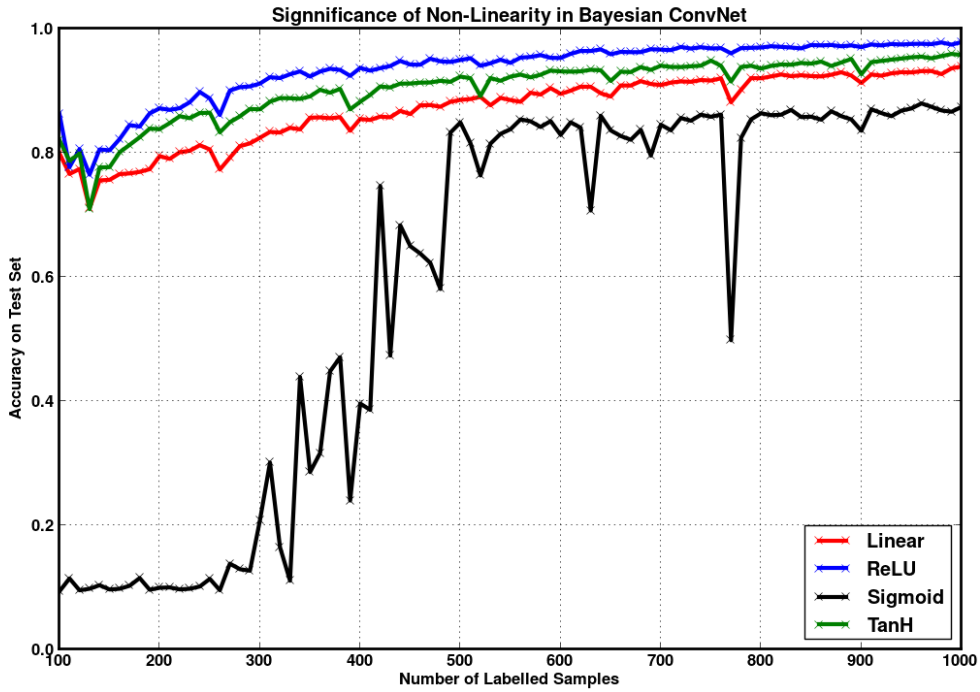


Fig. 3.11 Significance of different non-linearity in the CNN architecture, corresponding to different GP covariance functions in the Bayesian CNN architecture, using Dropout BALD acquisition function

Figure 3.11 illustrates the significance of using different activation functions or non-linearities in the Bayesian CNN implementation. The result shows the importance of using ReLU activation functions in CNN model compared to using the sigmoid activations. The different activation functions would give different uncertainty estimates from the Bayesian CNN model, since each GP covarince function has a one-to-one correspondence with NN non-linearities. Figure 3.11 illustrates that using a sigmoid activation function can make the active learning algorithm perform very poorly. This is because sigmoid is not a good activation function for use in deep neural networks, unlike ReLU and TanH activations. Our result here shows that a good uncertainty estimate obtained from a Bayesian CNN model can significantly impact the performance of our active learning algorithms. Furthermore, figure 3.11 justifies that using a ReLU activation function is better to obtain uncertainty esitmates from a Bayesian CNN model compared to using sigmoid activations, which offer very poor uncertainty estimates from CNNs for active learning.

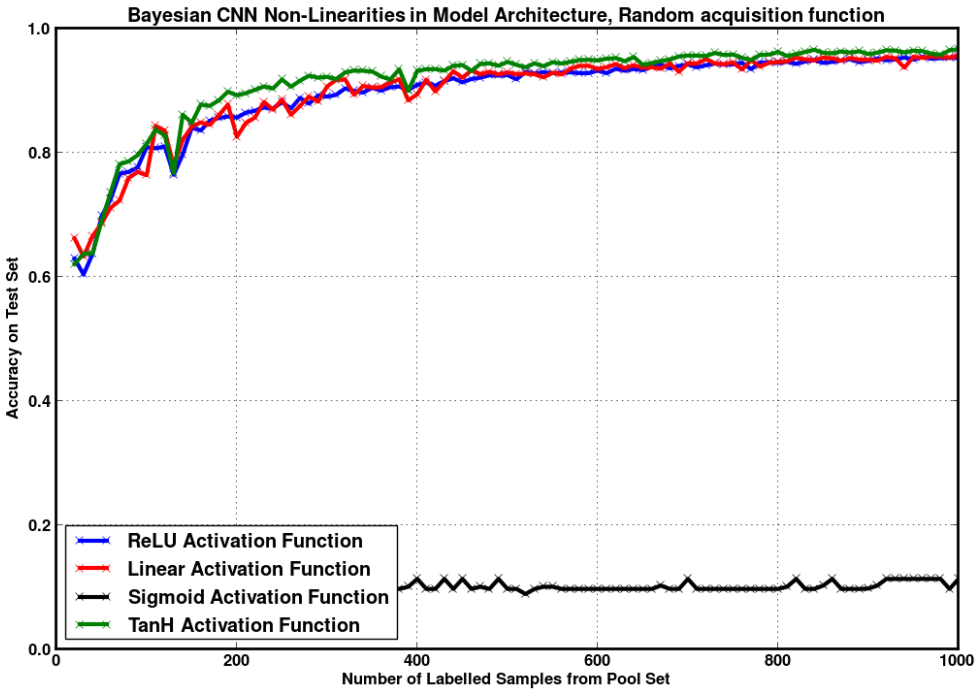


Fig. 3.12 Comparing Bayesian CNN model non-linearities on the Random acquisition function

We further compare the different Bayesian CNN non-linearities on the random acquisition function. Figure 3.12 again illustrates that the ReLU and TanH non-linearities mostly outperforms, while the sigmoid activation function performs poorly. This further justifies the poor uncertainty estimate that we get from the sigmoid activation, which is comparable to a poorly chosen covariance function for the equivalent GP.

Comparing figures 3.12 and 3.11, it is interesting to note the significance of using the BALD function compared to random acquisitions. Using Dropout BALD, for higher number of samples, the performance of the sigmoid model architecture improves, whereas for random acquisition it always performs poorly. The results in this section further justifies that for deep learning models, we cannot use a sigmoid activation function at the top NN layer of the CNN model. Our results not only illustrate the significance of Dropout BALD, but also demonstrates the importance of choosing the appropriate model non-linearities for obtaining good uncertainty estimates from the predictive distribution of the Bayesian CNN (based on choosing ReLU versus Sigmoid activations) for active learning.

3.5.3 Bayesian CNN Model Architectures

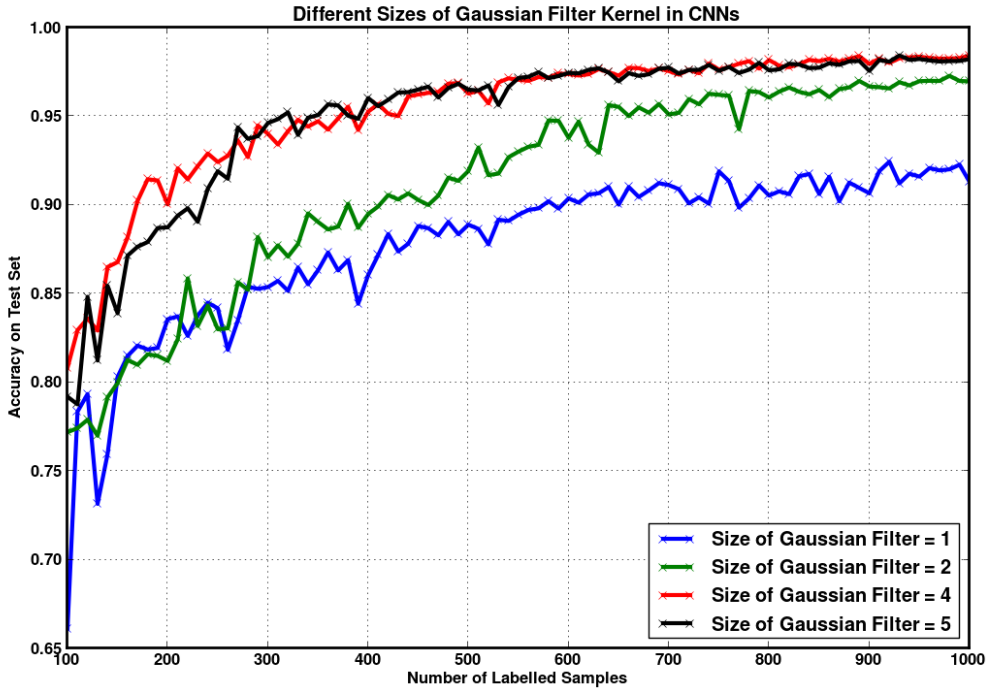


Fig. 3.13 Signifance of different non-linearity in the CNN architecture, corresponding to different GP covariance functions in the Bayesian approximation of Dropout

We then evaluated different model architectures for the Bayesian CNN LeNet5 architecture. We evaluated different sizes of the Gaussian kernel of the CNN to see how modelling of the distribution over the kernels (ie filters) is affected for different sizes. Furthermore, we experimented with different number of hidden units in the top NN layer of the Bayesian CNN model. These are tunable parameters which affects the performance of the active learning algorithm. For future work, these parameters can also be fine-tuned using Bayesian optimization [34]. Our experimental results in figure 3.13 shows that by fine-tuning the CNN model configutations, we can further improve the predictive performance of our active learners for images.

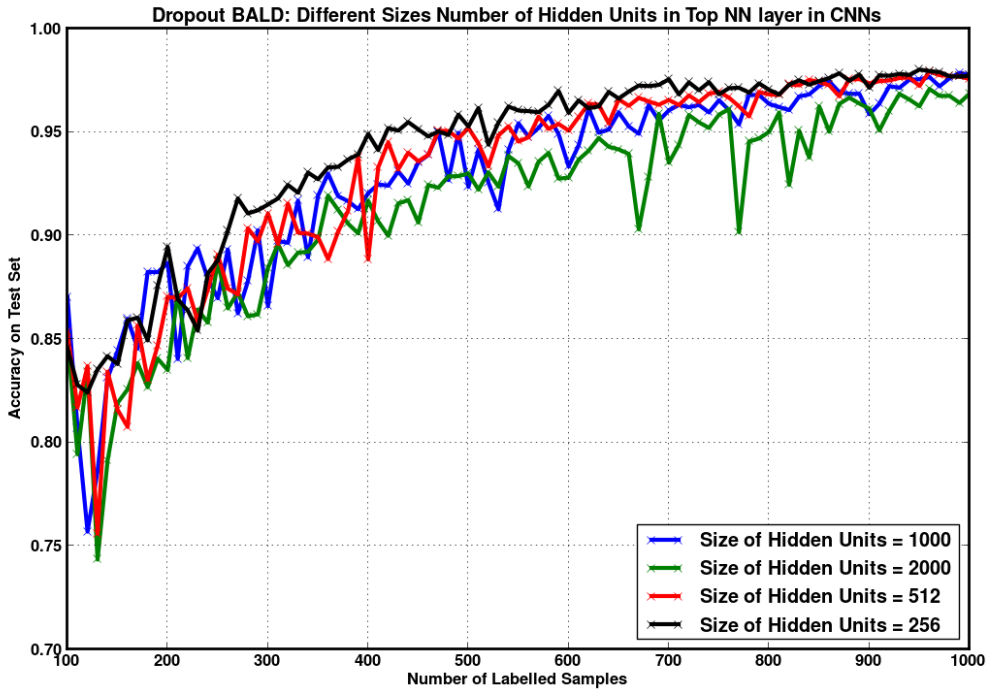


Fig. 3.14 Signifance of different non-linearity in the CNN architecture - influence of the number of hidden units in top NN layer in a CNN

Figure 3.14 then shows the significance of the number of hidden units in the top NN layer of the Bayesian CNN model. From figure 3.14 we can conclude that the number of hidden units perhaps does not play an important role in varying the uncertainty estimates from a Bayesian CNN model. Again, this parameter can be fine-tuned by using Bayesian optimization [34].

3.5.4 Discussion

Figures 3.11 and 3.12 shows the significance of the non-linear units in the output of a CNN, which approximates to different GP covariance functions. Hence, the non-linear units changes the uncertainty estimates obtained from our Bayesian CNN model which further affects the performance of the active learners. Additionally, figure 3.13 shows the effect in the performance of the active learning algorithm for different sizes of kernels. Different kernel filters using in CNNs when combined with the Bayesian approximation to dropout can give different uncertainty estimates. We also evaluated the significance of using different number of hidden units at the top NN layer of a CNN architecture. It is well known that an infinite number of hidden units corresponds to GP approximation and so we evaluate the significance of increasing the total number of hidden units at the top layer of our CNN model

architecture. Different number of hidden units also corresponds to different GP covariance functions and hence different uncertainty estimates over image classification.

3.6 Significance of Computation Time in Active Learning

One difficulty of performing active learning in a deep learning setting is that the model needs to be fitted with every new query point acquisition. In other words, every time a query is made from the pool set, the model needs to be fitted again. In the deep learning setting, this maybe difficult because such models are often highly prone to overfitting, especially when using a small dataset. In this section, we investigate the significance of query rate. Instead of querying only one point at a time from the pool set, we evaluate the trade-offs of querying more than one point at a time, to avoid the expensive model re-training process at every iteration.

3.6.1 Experimental Results

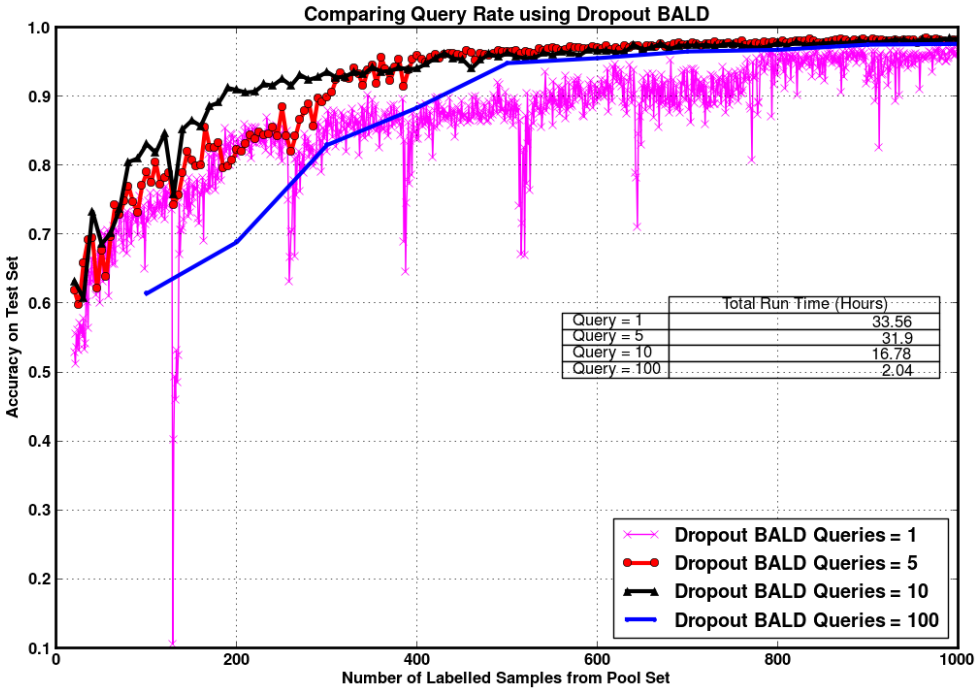


Fig. 3.15 Significance of Query Rate and Computation Time for active learning in deep learning

The experimental results in figure 3.15 shows that the query rate, even though varies the accuracy rate initially, eventually the same level of predictive performance is reached. Our results demonstrate the importance of the number of queries to be made at each active learning acquisition iteration. Furthermore, the table included in the figure shows the total computation time for each of the experiments. From figure 3.15 we can conclude that by querying more points at every iteration, we can improve the rate at which the accuracy increases, while also lowering the total computation required. In other words, by querying a higher number of points every iteration, we can reduce the total number of times the CNN models need to be re-trained, which is useful in our active learning in deep learning framework.

3.6.2 Discussion

Figure 3.15 illustrates the significance of query rate in active learning, which is importance especially considering this setting in the deep learning framework. Deep learning models are known to require large amounts of training data, and so querying only one point at a time, and re-training a deep model for every acquisition iteration maybe computationally quite expensive. In figure 3.15 we therefore illustrate that, instead of querying only one point, ie choosing the most informative point, we can instead choose 5 or 10 most informative points at a time that the model is highly uncertain about. In figure 3.15 note how the accuracy rate of the active learner depends on the query rate. Our results show that, instead of querying only point at time, it may perhaps be better to query 5 or 10 points at a time.

Another reason why querying only one point and adding this point to train a deep model is perhaps less useful because this single point added to the deep network gets smoothed out in the loss function. In other words, the addition of a single point does not bring a significant effect in the training of the network, unless these new additional points are highly weighted compared to the previous points.

In figure 3.15, we also make a comparison of the total computation time for each of the experiments, depending on the query rate. Comparing $Query = 5$ and $Query = 10$, we find that the later achieves a higher accuracy rate, while also having a lower computation time of almost $32 hours$. In comparison, $Query = 1$ and $Query = 5$ takes almost double the computation time (more than $30 hours$) while still not achieving a high enough accuracy rate for the active learner. Our results also demonstrate that querying 100 points at a time is not useful since we are selecting too many points that the model is not confident about. In other words, $Q = 100$ means that we are not critically querying the most informative points

from the pool set, which is also justified by its lower accuracy rate. From our experiments demonstrated in figure 3.15, we therefore show that using a query rate of 10 is a good balance in trading off accuracy rate and computation time. To re-emphasize, balancing this trade-off is important specially considering active learners using deep models such as Bayesian CNNs as classifiers.

3.7 Approximate Bayesian Neural Networks and Deep Gaussian Processes

In section 2.4.1 we discussed that while there exists other methods such as deep Gaussian Processes (DGPs) and approximate Bayesian methods for training neural networks, the dropout training in neural networks as approximate Bayesian inference tool can only be suitably applied for an extension in CNNs compared to other methods [6]. We repeat that, even though other methods such as variational methods, expectation propagation in DGPs and probabilistic backpropgation can give suitable uncertainty estimates for complex models, these methods have not yet been shown to be suitably applied to CNN models. These methods have been shown to give good uncertainty measures in regression tasks, and some have been shown to work well for low dimensional classification tasks. For example, even though the approximate expecation propagation scheme for DGPs [27] can give good uncertainty estimates in regression task, it cannot be suitably applied to high dimensional classification tasks at all, especially considering inputs such as images for CNN models.

In this section, we compare the methods discussed in section 2.4.1 with the MC dropout scheme [9] in an active learning regression setting. Our experimental results in this secton are to demonstrate that we can rely on the dropout uncertainty estimates for use in active learning, tested in a regression setting. We compare how good the uncertainty estimates are from each of these methods to be able to perform active learning. Even though the main focus of our work is for active learning in image data, here we demonstrate these results only to show that the uncertainty estimates from dropout are reliable similar to those from probabilistic backpropagation or DGPs. Note that the results here are not to find the best model that gives the best uncertainty estimate for active learning, but to demonstrate that the uncertainty estimates from MC dropout in NNs can be relied up. We demonstrate this through a regression task, using the Boston Housing dataset only. Further from this, one interesting direction for future work would be extend models such as probabilistic backpropagation [26]

for use in CNNs to obtain a different Bayesian CNN implementation, or perhaps to be able to use Deep GPs for higher dimensional inputs such as images.

3.7.1 Experimental Results

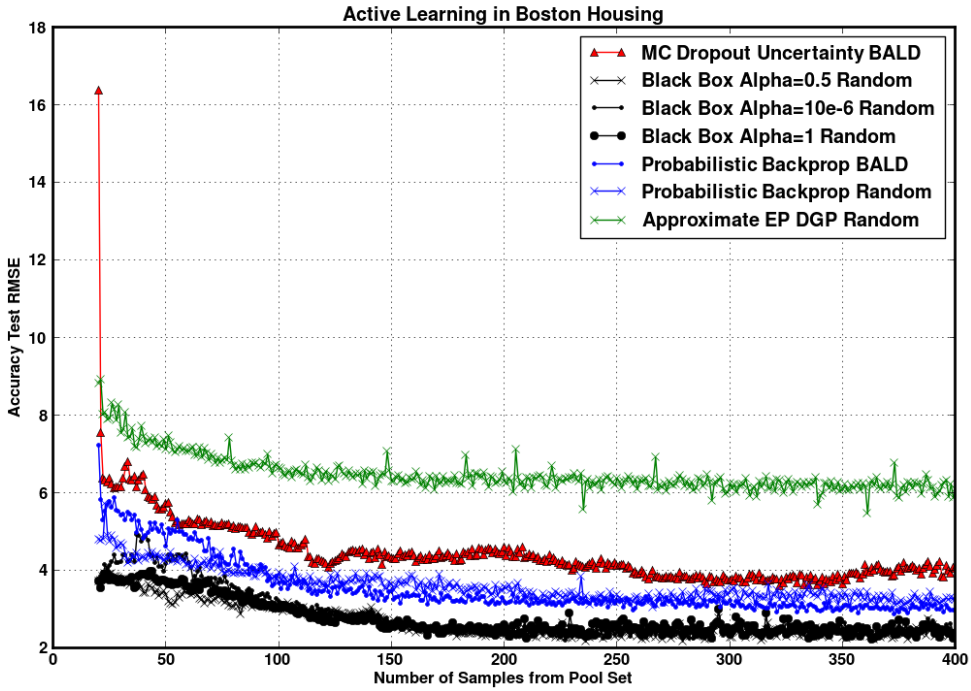


Fig. 3.16 Comparison of dropout uncertainty with probabilistic backpropagaton, Black-Box Alpha divergence and Deep Gaussian Process in an active learning regression task

We compare our results with several other methods for obtaining uncertainty estimates. For example, we use the Black-Box alpha implementation with different values of α [35] parameters on an active learning setting. We further compare our results with the probabilistic backpropagation algorithm [26] and the Deep Gaussian processe based approach based on using approximate EP [27].

Figure 3.16 compares the different methods discussed above in an active learning regression task. We illustrate these results using the Boston Housing dataset, starting with only 20 training datapoints and querying upto 400 training samples. We used a given configuration for the dropout uncertainty NN model, and compared it with different α values in Black-Box

alpha, probabilistic backpropagation [26] and a readily available implementation of the Deep Gaussian Process [27].

3.7.2 Discussion

The results from this section illustrates that the dropout uncertainty estimates can be relied upon for extension to classification tasks. Even though figure 3.16 shows that the BB-α outperforms all the other methods, this method has not yet been shown to perform well on classification tasks, and yet to demonstrate good performance for high dimensional inputs such as images. From figure 3.16, we want to justify that even though the dropout uncertainty estimate may not be as good as BB-α for this specific active learning regression setting, the MC dropout Bayesian approximation is the only available method to be easily extendable to CNNs, and therefore can be used for active learning in image classification tasks using Bayesian CNNs, while also avoiding overfitting for the small data regime.

3.8 Combining Active and Semi-Supervised Learning

In this section, we compare our dropout uncertainty acquisition functions with the approach from [31] that combines active learning and semi-supervised learning methods using Gaussian random fields and harmonic energy functions discussed previously in section 2.5. We implemented the approach from [31] based on constructing Gaussian random fields with raw image features using a RBF kernel in keras, while also using a CNN as the model classifier. We compare our results in a binary classification setting. We compare binary classification experiments comparing digits 2 and 8, and digits 3 and 8, to illustrate the difference in performance between an active learning method, and a method that combines active learning with semi-supervised learning. The semi-supervised learning approach using Gaussian random fields was previously implemented in [31] using a Bayes risk classifier. We compare this scheme with our proposed active learning algorithms, but only considering a binary classifier for image classification tasks.

3.8.1 Experimental Results

Figure 3.17 shows the comparison of our proposed active learning algorithms with an approach that combines active learning with graph-based semi-supervised learning. Figure 3.17 shows results for a binary classification task.

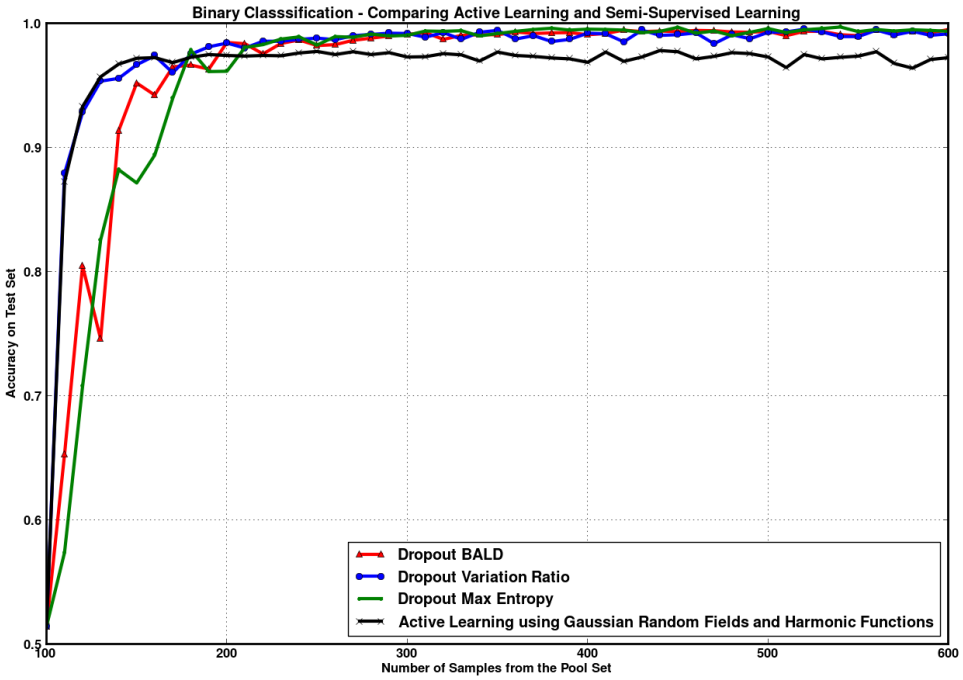


Fig. 3.17 Comparing dropout uncertainty active learning algorithms with graph-based semi-supervised learning algorithm using Gaussian random fields and Harmonic functions. Comparison of digits 2 and 8

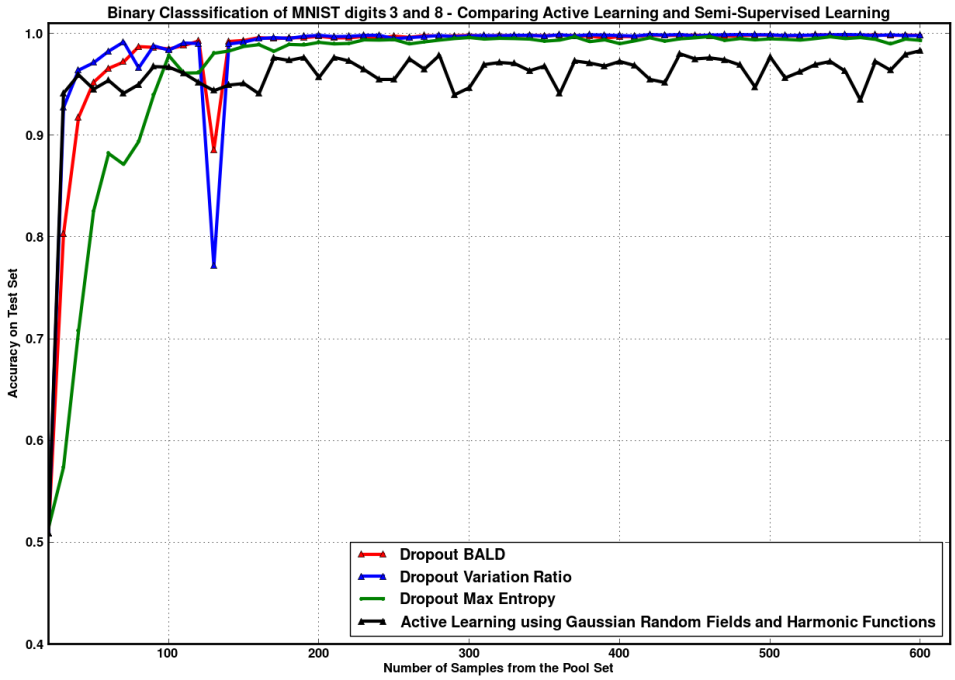


Fig. 3.18 Comparing dropout uncertainty active learning algorithms with graph-based semi-supervised learning algorithm using Gaussian random fields and Harmonic functions. Comparison of digits 3 and 8

Furthermore, figure 3.18 illustrates another experiment comparing digits 3 and 8. Our results in this section shows that in both figures 3.18 and 3.17, our dropout active learning algorithms outperforms the Gaussian random field based active learning approach, both implemented on a LeNet5 CNN classifier.

3.8.2 Discussion

The experimental results from figure 3.17 and 3.18 demonstrates that our dropout uncertainty active learning algorithms outperforms the approach based on constructing graphs using semi-supervised learning, even though the latter method is implemented with a CNN classifier. From both figures 3.17 and 3.18, even though the semi-supervised learning based approach has an initial high accuracy rate, it eventually performs poorly compared to proposed active learning algorithms. In addition, our active learner Dropout Variation Ratio also has a similar accuracy rate compared to the Gaussian random field based approach.

3.9 Comparison with Semi-Supervised Learning

In this section, we summarise all the results of our proposed active learners on the MNIST image classification task. The experimental results below shows the high classification accuracy that our proposed active learning algorithms can achieve using a Bayesian CNN model trained with few labelled samples. Table 3.1 below summarises our results in terms of the maximum accuracy that can be achieved on MNIST classification task with 10,000 test samples. We show our results for 100, 1000 and 3000 labelled training samples and show how the test set accuracy can be improved as we query more points from the pool set based on the information gain.

Table 3.1 Summary of Active Learning Experimental Results

Test Accuracy Results on MNIST for 100, 1000 and 3000 labelled training samples

Test accuracy % on 10,000 test samples with number of used training labels	100	1000	3000
Dropout BALD	85.69	98.43	98.84
Dropout Variation Ratio	87.89	98.36	98.87
Dropout Maximum Entropy	89.55	98.26	98.84
Dropout Least Confident	89.4	97.86	98.87
Dropout Bayes Segnet	83.52	95.87	97.19
Random Acquisition	84.86	94.95	97.31
Best vs Second Best (Max Margin)	79.25	83.95	82.77
Maximum Entropy	73.86	97.70	98.20

Our experimental results show that using only 1000 labelled samples for training, testing on 10,000 samples, we can achieve a high enough classification accuracy, and the increase in the number of samples from 1000 to 3000 does not bring a significant improvement. This demonstrates that using active learning with the Bayesian CNN, we can train MNIST image classification models with only 1000 training samples in order to achive a very high test accuracy. From table 3.1 below, for 1000 labelled samples, our proposed Dropout BALD active learning algorithm achieves the best performing classification accuracy of 98.43%.

We further compare our active learning algorithms with other proposed methods mainly based on semi-supervised learning schemes. We re-emphase that our work is the first of its kind to use active learning in a deep learning framework to achieve data-efficiency in image processing tasks. We therefore cannot compare our results with other state of the art

active learning algorithms. The method most similar to us is based on using semi-supervised learning. Table 3.2 below further summarises the results. Table 3.2 shows that our Dropout BALD achieves a test error of 1.57% which is close to the current state of the art on MNIST (using semi-supervised learning) of a test error of 0.84%. From table 3.2, we demonstrate that our proposed methods can achieve data-efficiency which is quite close to the current state of the art. We repeat here that our focus is not to achieve the state of the art performance on MNIST, but to demonstrate that it is possible to use active learning in the deep learning framework which had not been done before. Table 3.2 illustrates that using Bayesian CNN implementation on MNIST, we can perform active learning in these settings and compare our results with semi-supervised learning methods. One important thing to remember is that, using active learning we only query few points at every acquisition iteration by estimating the predictive uncertainty over the pool points using test-time MC dropout. This is a very easy to implement and efficient approach to obtain predictive uncertainty over the pool set. In contrast, the semi-supervised learning methods compared here needs to take account of all the images from the pool set which is more expensive compared to simply applying test-time dropout. Although these approaches included in table 3.2 is not directly comparable with our results, it is the closest approach to compare in the framework of data-efficiency in deep learning.

Table 3.2 Comparison between Active Learning and Semi-Supervised Learning methods

Test Error Results on MNIST for 1000 labelled training samples

Test error % on 10,000 samples with number of used training labels	1000
Semi-sup. Embedding (Weston et al., 2012)	5.73
MTC (Rifai et al., 2011)	3.64
Pseudo-label (Lee, 2013)	3.46
AtlasRBF (Pitelis et al.,2014)	3.68
Semi-Supervised with GAN (Odena et al., 2016)	3.60
DGN (Kingma et al., 2014)	2.40
Virtual Adversarial (Miyato et al., 2015)	**1.32**
SSL with Ladder Networks (Rasmus et al., 2015)	**0.84**
Dropout BALD	**1.57**
Dropout Variation Ratio	**1.64**
Dropout Maximum Entropy	**1.74**
Dropout Least Confident	**2.14**
Dropout Bayes Segnet	**4.13**

As discussed above, the experimental results in 3.2 shows that using our proposed active learning method in the deep learning framework for MNIST image classification task, we can achieve similar levels of performance as that achieved through the use of semi-supervised learning. More importantly, our algorithm can outperform the approach based on using deep generative models using an variation auto-encoder [36], and the more recent approaches based on combining semi-supervised learning with generative adversarial networks [37].

3.10 Summary of Experimental Results

In this chapter, we have presented our experimental results using the proposed active learning algorithms based on dropout model uncertainty obtained from Bayesian CNN. Our results illustrate that the Bayesian CNN model does not overfit in the active learning image classification setting. We compared our proposed methods with several baseline acquisition functions typically used in active learning to demonstrate that our method outperforms on the MNIST dataset by obtaining model predictive uncertainty, which is useful for querying the most informative points. Furthermore, we demonstrated the importance of uncertainty estimates in active learning by comparing our proposed acquisition functions with softmax

output of a CNN, and by considering several CNN model architectures and non-linearities which corresponds to different GP covariance functions for uncertainty estimates. Since we are the first to consider active learning in a deep learning framework, our results further demonstrated the importance of computation time in active learning, and we showed that instead of querying only one point at a time from the pool set, it is more computationally efficient to query upto 10 image points from the pool set. In order to illustrate that our uncertainty estimation from dropout is reliable, we further compared our results in a simple active learning regression task, comparing our method with other approximate Bayesian and DGPs which can also give model uncertainty. However, we showed that only the MC dropout model Bayesian approximation can be suitably extended to CNN models unlike other methods, when considering active learning for image data. We further compared our proposed active learning method with a graph-based semi-supervised learning scheme which combines active learning on a binary classification task. Our results show that simply using active learning, it is more efficient to improve test accuracy, compared to considering semi-supervised learning approaches. Finally, we showed that using our proposed active learning algorithms, we can achieve data-efficiency in deep learning, and achieve a test set accuracy on MNIST data which is very close to the current state-of-the-art. Our method also outperforms several other recent approaches which are based on semi-supervised learning.

Chapter 4

Conclusions

4.1 Summary and Discussion

In this thesis, we introduced the framework of using active learning for image data using Bayesian convolutional neural networks. We propose the use of active learning in a deep learning framework for image classification task which has not been explored before. This was mainly because CNNs in deep learning were known to require large amounts of training data. Therefore, previously active learning could not be used as a suitable framework since the goal of active learning was to reduce the number of training data required while maining similar levels of classification performance.

In our work, we build on the recently proposed framework of Bayesian convolutional neural networks. Recent work showed that dropout training in neural networks can be cast as approximate Bayesian inference in neural networks [9]. This was suitably extended for proposing Bayesian convolutional neural networks [6], illustrating the significance that Bayesian ConvNets can be trained with small amounts of training data for images. Further from this, it was shown from [9] that model confidence and uncertainty can be represented in a deep learning framework, by using average stochastic forward passes of dropout at test-time. The predictive distribution obtained by performing Monte-Carlo estimates of dropout can be used as a measure of uncertainty for image classification tasks.

We combine the framework of Bayesian approximation to dropout in CNN models to obtain uncertainty estimates from a predictive distribution with information theoretic active learning. We use the model uncertainty from Bayesian CNNs to propose information thoertic entropy based measures for active learning. In this work, we have proposed several new acquisition functions that can be used in a deep learning setting when using active

learning. Furthermore, our work have shown that by capturing model uncertainty from the Bayesian approximation to dropout, we can query the most informative points with which a deep model such as a CNN can be trained to achieve close to state of the art results on the MNIST dataset. Our experimental results have shown that our proposed acquisition functions can easily outperform the acquisition functions typically used in active learning. We have also justified the importance of a good uncertainty estimate that is required for active learning.

The work in this thesis further illustrates the significance of using active learning in a deep learning framework for image classification task. Our work is the first to propose active learning using Bayesian CNN classifiers for image datasets. We illustrate state-of-the-art predictive accuracy measures on the MNIST dataset using the active learning framework. Our framework is the first to propose that active learning can be performed with CNN models for images. We demonstrate that using very few labeled samples for training, we can achieve data-efficiency in image classification tasks, by querying the most informative image points by following our proposed acquisition functions. By comparing our results with semi-supervised learning methods, we can achieve a test set performance which is close to the current state of the art, but using an active learning algorithm in the deep learning framework.

4.2 Future Work

This thesis work can open up many opportunities for the use of active learning in the deep learning framework, towards the overall goal of achieving data-efficiency in deep learning which is an open research problem. In our work, we only considered the use of dropout as Bayesian approximation to represent model uncertainty for active learning. Although other methods for representing uncertainty also exists, they have not yet been shown to perform well on CNN models. One useful future research direction will be to extend other approximate Bayesian NN methods onto CNN models to provide a different interpretation of uncertainty estimates over classification tasks. For future work, it would be useful to come with a more calibrated uncertainty estimate to further improve the performance and data-efficiency of active learning algorithms.

Additionally, it would be interesting to see if these methods can be applied to more real world applications, for example considering health care data such as Brain MRI scans where labelled data is scarce. These algorithms, if efficiently implemented in real-world data

settings, then it can have a major impact towards bringing data-efficiency in deep learning. Furthermore, one possible future direction is to consider video processing tasks using active learning, or other similar frameworks in computer vision. A lot of recent work considered the task of image and video caption generation, where these models use attention based models. It would be interesting to see if active learning can further help towards bringing data-efficiency in these caption generation systems.

For future research in this framework of applying active learning in deep learning context, one obvious direction would be to come up with more efficient active learning acquisition functions. Since active learning involves repeated training of the deep model, it would be interesting if this repeated training can be avoided to save computation time and resource. This is also one reason why active learning was not previously considered with CNN models, since repeated training of CNN models is very expensive. However, for future work, we would want to come up with a clever way of solving this problem to make active learning more widely used for deep learning applications such as in computer vision, natural language processing and speech recognition. Additionally, data-efficiency is still an open research problem in deep reinforcement learning frameworks, where agents need to be trained with large amounts of training data (trajectories). It would be interesting to consider the use of active learning in deep reinforcement learning settings, where agents can be trained with the most informative trajectories from its experience.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.

[2] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition.

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.7298594. URL http://dx.doi.org/10.1109/CVPR.2015.7298594.

[4] David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472, May 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.3.448. URL http://dx.doi.org/10.1162/neco.1992.4.3.448.

[5] Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996. ISBN 0387947248.

[6] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *CoRR*, abs/1506.02158, 2015. URL http://arxiv.org/abs/1506.02158.

[7] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models, 1995.

[8] David J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992. doi: 10.1162/neco.1992.4.4.590. URL http://dx.doi.org/10.1162/neco.1992.4.4.590.

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *CoRR*, abs/1506.02142, 2015. URL http://arxiv.org/abs/1506.02142.

[10] Andreas C. Damianou and Neil D. Lawrence. Deep gaussian processes. *CoRR*, abs/1211.0358, 2012. URL http://arxiv.org/abs/1211.0358.

[11] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001. URL http://www.ai.mit.edu/projects/jmlr/papers/volume2/tong01a/abstract.html.

[12] Simon Tong and Edward Y. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the 9th ACM International Conference on Multimedia 2001, Ottawa, Ontario, Canada, September 30 - October 5, 2001*, pages 107–118, 2001. URL http://portal.acm.org/citation.cfm?id=500141.500159.

[13] Gökhan Tür, Dilek Z. Hakkani-Tür, and Robert E. Schapire. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186, 2005. doi: 10.1016/j.specom.2004.08.002. URL http://dx.doi.org/10.1016/j.specom.2004.08.002.

[14] Rong Jin and Luo Si. A bayesian approach toward active learning for collaborative filtering. In *UAI '04, Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence, Banff, Canada, July 7-11, 2004*, pages 278–285, 2004. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1119&proceeding_id=20.

[15] Claude E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[16] D. V. Lindley. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, 27(4):986–1005, 1956. ISSN 0003-4851. doi: 10.1214/aoms/1177728069.

[17] Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *CoRR*, abs/1112.5745, 2011. URL http://arxiv.org/abs/1112.5745.

[18] Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *CVPR*, pages 2372–2379. IEEE Computer Society, 2009. ISBN 978-1-4244-3992-8. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2009.html#JoshiPP09.

[19] Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, pages 5–13, New York, NY, USA, 1993. ACM. ISBN 0-89791-611-5. doi: 10.1145/168304.168306. URL http://doi.acm.org/10.1145/168304.168306.

[20] D. Barber and C.M. Bishop. Ensemble learning in bayesian neural networks. In *Neural Networks and Machine Learning*, pages 215–237. Springer, 1998. URL ../publications/barber-bishop-ensemble-NATO98.pdf.

[21] Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 2348–2356, 2011. URL http://papers.nips.cc/paper/4329-practical-variational-inference-for-neural-networks.

[22] John William Paisley, David M. Blei, and Michael I. Jordan. Variational bayesian inference with stochastic search. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012. URL http://icml.cc/discuss/2012/687.html.

[23] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL http://arxiv.org/abs/1312.6114.

[24] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1278–1286, 2014. URL http://jmlr.org/proceedings/papers/v32/rezende14.html.

[25] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *CoRR*, abs/1505.05424, 2015. URL http://arxiv.org/abs/1505.05424.

[26] José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1861–1869, 2015. URL http://jmlr.org/proceedings/papers/v37/hernandez-lobatoc15.html.

[27] Thang D. Bui, Daniel Hernández-Lobato, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E. Turner. Deep gaussian processes for regression using approximate expectation propagation. *CoRR*, abs/1602.04133, 2016. URL http://arxiv.org/abs/1602.04133.

[28] Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 2372–2379, 2009. doi: 10.1109/CVPRW.2009.5206627. URL http://dx.doi.org/10.1109/CVPRW.2009.5206627.

[29] Alex Holub, Pietro Perona, and Michael C. Burl. Entropy-based active learning for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2008, Anchorage, AK, USA, 23-28 June, 2008*, pages 1–8, 2008. doi: 10.1109/CVPRW.2008.4563068. URL http://dx.doi.org/10.1109/CVPRW.2008.4563068.

[30] Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 859–866, 2013. doi: 10.1109/CVPR.2013.116. URL http://dx.doi.org/10.1109/CVPR.2013.116.

[31] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65, 2003.

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

[33] François Chollet. keras. https://github.com/fchollet/keras, 2015.

[34] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing*

Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States., pages 2960–2968, 2012. URL http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.

[35] José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Thang D. Bui, Daniel Hernández-Lobato, and Richard E. Turner. Black-box alpha divergence minimization. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1511–1520, 2016. URL http://jmlr.org/proceedings/papers/v48/hernandez-lobatob16.html.

[36] Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3581–3589, 2014. URL http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.

[37] Augustus Odena. Semi-supervised learning with generative adversarial networks. *CoRR*, abs/1606.01583, 2016. URL http://arxiv.org/abs/1606.01583.
[utf8]inputenc [english]babel hyperref natbib amsmath,amssymb E