Deep Bayesian Active Learning with Image Data

Yarin Gal [1 2] **Riashat Islam** [1] **Zoubin Ghahramani** [1 3]

Abstract

Even though active learning forms an important pillar of machine learning, deep learning tools are not prevalent within it. Deep learning poses several difficulties when used in an active learning setting. First, active learning (AL) methods generally rely on being able to learn and update models from small amounts of data. Recent advances in deep learning, on the other hand, are notorious for their dependence on large amounts of data. Second, many AL acquisition functions rely on model uncertainty, yet deep learning methods rarely represent such model uncertainty. In this paper we combine recent advances in Bayesian deep learning into the active learning framework in a practical way. We develop an active learning framework for high dimensional data, a task which has been extremely challenging so far, with very sparse existing literature. Taking advantage of specialised models such as Bayesian convolutional neural networks, we demonstrate our active learning techniques with image data, obtaining a significant improvement on existing active learning approaches. We demonstrate this on both the MNIST dataset, as well as for skin cancer diagnosis from lesion images (ISIC2016 task).

1. Introduction

A big challenge in many machine learning applications is obtaining labelled data. This can be a long, laborious, and costly process, often making the deployment of ML systems uneconomical. A framework where a system could learn from small amounts of data, and choose by itself what data it would like the user to label, would make machine learning much more widely applicable. Such frameworks for learning are referred to as *active learning* (Cohn et al., 1996) (also known as "experiment design" in the statistics literature), and have been used successfully in fields such as medical diagnosis, microbiology, and manufacturing (Tong, 2001). In active learning, a model is trained on a small amount of data (the initial training set), and an *acquisition function* (often based on the model's *uncertainty*) decides which data points to ask an external *oracle* for a label. The acquisition function selects one or more points from a *pool* of unlabelled data points, with the pool points lying outside of the training set. An oracle (often a human expert) labels the selected data points, these are added to the training set, and a new model is trained on the updated training set. This process is then repeated, with the training set increasing in size over time. The advantage of such systems is that they often result in dramatic reductions in the amount of labelling required to train an ML system (and therefore cost and time).

Even though existing techniques for active learning have proven themselves useful in a variety of tasks, a major remaining challenge in active learning is its lack of scalability to high-dimensional data (Tong, 2001). This data appears often in image form, with a physician classifying MRI scans to diagnose Alzheimer's for example (Marcus et al., 2010), or an expert clinician diagnosing skin cancer from dermoscopic lesion images. To perform active learning, a model has to be able to learn from small amounts of data and represent its uncertainty over unseen data. This severely restricts the class of models that can be used within the active learning framework. As a result most approaches to active learning have focused on low dimensional problems (Tong, 2001; Hernandez-Lobato & Adams, 2015), with only a handful of exceptions (Zhu et al., 2003; Holub et al., 2008; Joshi et al., 2009) relying on kernel or graph-based approaches to handle high-dimensional data.

In recent years, with the increased availability of data in *some* domains, attention within the machine learning community has shifted from small data problems to big data problems (Sundermeyer et al., 2012; Krizhevsky et al., 2012; Kalchbrenner & Blunsom, 2013; Sutskever et al., 2014). And with the increased interest in big data problems, new tools were developed and existing tools were refined for handling high dimensional data within such regimes. Deep learning, and convolutional neural networks (CNNs) (Rumelhart et al., 1985; LeCun et al., 1989) in particular, are an example of such tools. Originally developed in 1989 to parse handwritten zip codes, these tools have flourished and were adapted to a point where a CNN is able to beat a human on object recognition tasks (given enough training data)

[1] University of Cambridge, UK [2] The Alan Turing Institute, UK [3] Uber AI Labs. Correspondence to: Yarin Gal <yg279@cam.ac.uk>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

(He et al., 2015). New techniques such as dropout (Hinton et al., 2012; Srivastava et al., 2014) are used extensively to regularise these huge models, which often contain millions of parameters (Jozefowicz et al., 2016). But even though active learning forms an important pillar of machine learning, deep learning tools are not prevalent within it. Deep learning poses several difficulties when used in an active learning setting. First, we have to be able to handle small amounts of data. Recent advances in deep learning, on the other hand, are notorious for their dependence on large amounts of data (Krizhevsky et al., 2012). Second, many AL acquisition functions rely on model uncertainty. But in deep learning we rarely represent such model uncertainty.

Relying on Bayesian approaches to deep learning, in this paper we combine recent advances in Bayesian deep learning into the active learning framework in a practical way. We develop an active learning framework for high dimensional data, a task which has been extremely challenging so far with very sparse existing literature from the past 15 years (Zhu et al., 2003; Li & Guo, 2013; Holub et al., 2008; Joshi et al., 2009). Taking advantage of specialised models such as Bayesian convolutional neural networks (BCNNs) (Gal & Ghahramani, 2016a;b), we demonstrate our active learning techniques with image data. Using a small model, our system is able to achieve 5% test error on MNIST with only 295 labelled images without relying on unlabelled data (in comparison, 835 labelled images are needed to achieve 5% test error using random sampling – requiring an expert to label more than twice as many images to achieve the same accuracy), and achieves 1.64% test error with 1000 labelled images. This is in comparison to 2.40% test error of DGN (Kingma et al., 2014) or 1.53% test error of the Ladder Network Γ-model (Rasmus et al., 2015), both semi-supervised learning techniques which additionally use the entire unlabelled training set. Finally, we study a real-world application by diagnosing melanoma (skin cancer) from a small number of lesion images by fine-tuning the VGG16 convolutional neural network (Simonyan & Zisserman, 2015) on the ISIC 2016 dataset (Gutman et al., 2016).

2. Related Research

Past attempts at active learning of image data have concentrated on kernel based methods. Using ideas from previous research in active learning of low dimensional data (Tong, 2001), Joshi et al. (2009) used "margin-based uncertainty" and extracted probabilistic outputs from support vector machines (SVM) (Cortes & Vapnik, 1995). They used linear, polynomial, and Radial Basis Function (RBF) kernels on the raw images, picking the kernel that gave best classification accuracy. Analogously to SVM approaches, Li & Guo (2013) used Gaussian processes (GPs) with RBF kernels to get model uncertainty. However Li & Guo (2013) fed low dimensional features (such as SIFT features) to their RBF kernel. Lastly, making use of unlabelled data as well, Zhu et al. (2003) acquire points using a Gaussian random field model, evaluating an RBF kernel over raw images. We compare to this last technique and explain it in more detail below.

Other related work includes semi-supervised learning of image data (Weston et al., 2012; Kingma et al., 2014; Rasmus et al., 2015). In semi-supervised learning a model is given a fixed set of labelled data, and a fixed set of unlabelled data. The model can use the unlabelled data to learn about the distribution of the inputs, in the hopes that this information will aid in learning from the small labelled set as well. Although the learning paradigm is fairly different from active learning, this research forms the closest modern literature to active learning of image data. We will compare to these techniques below as well, in section 5.4.

3. Bayesian Convolutional Neural Networks

In this paper we concentrate on high dimensional *image* data, and need a model able to represent prediction uncertainty on such data. Existing approaches such as (Zhu et al., 2003; Li & Guo, 2013; Joshi et al., 2009) rely on kernel methods, and feed image pairs through linear, polynomial, and RBF kernels to capture image similarity as an input to an SVM for example. In contrast, we rely on specialised models for image data, and in particular on convolutional neural networks (CNNs) (Rumelhart et al., 1985; LeCun et al., 1989). Unlike the kernels above, which cannot capture spatial information in the input image, CNNs are designed to use this spatial information, and have been used successfully to achieve state-of-the-art results (Krizhevsky et al., 2012). To perform active learning with image data we make use of the Bayesian equivalent of CNNs, proposed in (Gal & Ghahramani, 2016a)[1]. These Bayesian CNNs are CNNs with prior probability distributions placed over a set of model parameters $\boldsymbol{\omega} = \{W_1, ..., W_L\}$:

$$\boldsymbol{\omega} \sim p(\boldsymbol{\omega}),$$

with for example a standard Gaussian prior $p(\boldsymbol{\omega})$. We further define a likelihood model

$$p(y = c|\mathbf{x}, \boldsymbol{\omega}) = \text{softmax}(\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}))$$

for the case of classification, or a Gaussian likelihood for the case of regression, with $\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x})$ model output (with parameters $\boldsymbol{\omega}$).

To perform approximate inference in the Bayesian CNN model we make use of stochastic regularisation techniques such as dropout (Hinton et al., 2012; Srivastava et al., 2014), originally used to regularise these models. As shown in (Gal & Ghahramani, 2016b; Gal, 2016) dropout and various other stochastic regularisation techniques can be used to perform practical approximate inference in complex deep models. Inference is done by training a model with dropout

[1] As far as we are aware, there are no other tools in current literature that offer model uncertainty in specialised models for image data, which perform as well as CNNs.

before every weight layer, and by performing dropout at test time as well to sample from the approximate posterior (stochastic forward passes, referred to as *MC dropout*).

More formally, this approach is equivalent to performing approximate variational inference where we find a distribution $q_\theta^*(\boldsymbol{\omega})$ in a tractable family which minimises the Kullback-Leibler (KL) divergence to the true model posterior $p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})$ given a training set $\mathcal{D}_{\text{train}}$. Dropout can be interpreted as a variational Bayesian approximation, where the approximating distribution is a mixture of two Gaussians with small variances and the mean of one of the Gaussians is fixed at zero. The uncertainty in the weights induces prediction uncertainty by marginalising over the approximate posterior using Monte Carlo integration:

$$\begin{aligned} p(y=c|\mathbf{x},\mathcal{D}_{\text{train}}) &= \int p(y=c|\mathbf{x},\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})\mathrm{d}\boldsymbol{\omega} \\ &\approx \int p(y=c|\mathbf{x},\boldsymbol{\omega})q_\theta^*(\boldsymbol{\omega})\mathrm{d}\boldsymbol{\omega} \\ &\approx \frac{1}{T}\sum_{t=1}^{T} p(y=c|\mathbf{x},\widehat{\boldsymbol{\omega}}_t) \end{aligned}$$

with $\widehat{\boldsymbol{\omega}}_t \sim q_\theta^*(\boldsymbol{\omega})$, where $q_\theta(\boldsymbol{\omega})$ is the Dropout distribution (Gal, 2016).

Bayesian CNNs work well with small amounts of data (Gal & Ghahramani, 2016a), and possess uncertainty information that can be used with existing acquisition functions (Gal, 2016). Such acquisition functions for the case of classification are discussed next.

4. Acquisition Functions and their Approximations

Given a model $\mathcal{M}$, pool data $\mathcal{D}_{\text{pool}}$, and inputs $x \in \mathcal{D}_{\text{pool}}$, an acquisition function $a(x,\mathcal{M})$ is a function of x that the AL system uses to decide where to query next:

$$x^* = \text{argmax}_{x\in\mathcal{D}_{\text{pool}}} a(x,\mathcal{M}).$$

We next explore various acquisition functions appropriate for our image data setting, and develop tractable approximations for us to use with our Bayesian CNNs. In tasks involving regression we often use the predictive variance or a quantity derived from this for our acquisition function (although we still need to be careful to query from informative areas rather than querying noise). For example, we might look for images with high predictive variance and choose those to ask an expert to label – in the hope that these will decrease model uncertainty. However, many tasks involving image data are often phrased as classification problems. For classification, several acquisition functions are available:

1. Choose pool points that maximise the predictive entropy (*Max Entropy*, (Shannon, 1948))
   $$\mathbb{H}[y|\mathbf{x},\mathcal{D}_{\text{train}}] := -\sum_c p(y=c|\mathbf{x},\mathcal{D}_{\text{train}}) \log p(y=c|\mathbf{x},\mathcal{D}_{\text{train}}).$$

2. Choose pool points that are expected to maximise the information gained about the model parameters, i.e. maximise the mutual information between predictions and model posterior (*BALD*, (Houlsby et al., 2011))
   $$\mathbb{I}[y,\boldsymbol{\omega}|\mathbf{x},\mathcal{D}_{\text{train}}] = \mathbb{H}[y|\mathbf{x},\mathcal{D}_{\text{train}}] - \mathbb{E}_{p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})}\big[\mathbb{H}[y|\mathbf{x},\boldsymbol{\omega}]\big]$$
   with $\boldsymbol{\omega}$ the model parameters (here $\mathbb{H}[y|\mathbf{x},\boldsymbol{\omega}]$ is the entropy of y given model weights $\boldsymbol{\omega}$). Points that maximise this acquisition function are points on which the model is uncertain on average, but there exist model parameters that produce disagreeing predictions with high certainty. This is equivalent to points with high variance in the input to the softmax layer (the logits) – thus each stochastic forward pass through the model would have the highest probability assigned to a different class.

3. Maximise the *Variation Ratios* (Freeman, 1965)
   $$\text{variation-ratio}[\mathbf{x}] := 1 - \max_y p(y|x,\mathcal{D}_{\text{train}})$$
   Like *Max Entropy*, *Variation Ratios* measures lack of confidence.

4. Maximise mean standard deviation (*Mean STD*) (Kampffmeyer et al., 2016; Kendall et al., 2015)
   $$\sigma_c = \sqrt{\mathbb{E}_{q(\boldsymbol{\omega})}[p(y=c|\mathbf{x},\boldsymbol{\omega})^2] - \mathbb{E}_{q(\boldsymbol{\omega})}[p(y=c|\mathbf{x},\boldsymbol{\omega})]^2}$$
   $$\sigma(\mathbf{x}) = \frac{1}{C}\sum_c \sigma_c$$
   averaged over all c classes $\mathbf{x}$ can take. Compared to the above acquisition functions, this is more of an ad-hoc technique used in recent literature.

5. *Random* acquisition (baseline): $a(\mathbf{x}) = \text{unif}()$ with unif() a function returning a draw from a uniform distribution over the interval $[0,1]$. Using this acquisition function is equivalent to choosing a point uniformly at random from the pool.

These acquisition functions and their properties are discussed in more detail in (Gal, 2016, pp. 48–52).

We can approximate each of these acquisition functions using our approximate distribution $q_\theta^*(\boldsymbol{\omega})$. For BALD, for example, we can write the acquisition function as follows:

$$\begin{aligned} \mathbb{I}[y,\boldsymbol{\omega}|\mathbf{x},\mathcal{D}_{\text{train}}] &:= \mathbb{H}[y|\mathbf{x},\mathcal{D}_{\text{train}}] - \mathbb{E}_{p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})}\big[\mathbb{H}[y|\mathbf{x},\boldsymbol{\omega}]\big] \\ &= -\sum_c p(y=c|\mathbf{x},\mathcal{D}_{\text{train}}) \log p(y=c|\mathbf{x},\mathcal{D}_{\text{train}}) \\ &\quad + \mathbb{E}_{p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})}\Big[\sum_c p(y=c|\mathbf{x},\boldsymbol{\omega}) \log p(y=c|\mathbf{x},\boldsymbol{\omega})\Big], \end{aligned}$$

with c the possible classes y can take. $\mathbb{I}[y,\boldsymbol{\omega}|\mathbf{x},\mathcal{D}_{\text{train}}]$ can be approximated in our setting using the identity $p(y=c|\mathbf{x},\mathcal{D}_{\text{train}}) = \int p(y=c|\mathbf{x},\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})\mathrm{d}\boldsymbol{\omega}$:

$$\mathbb{I}[y,\boldsymbol{\omega}|\mathbf{x},\mathcal{D}_{\text{train}}] = -\sum_c \int p(y=c|\mathbf{x},\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})\mathrm{d}\boldsymbol{\omega}$$

$$\cdot \log \int p(y=c|\mathbf{x},\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})\mathrm{d}\boldsymbol{\omega}$$
$$+ \mathbb{E}_{p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})}\Big[\sum_c p(y=c|\mathbf{x},\boldsymbol{\omega})\log p(y=c|\mathbf{x},\boldsymbol{\omega})\Big].$$

Swapping the posterior $p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})$ with our approximate posterior $q_\theta^*(\boldsymbol{\omega})$, and through MC sampling, we then have:

$$\approx -\sum_c \int p(y=c|\mathbf{x},\boldsymbol{\omega})q_\theta^*(\boldsymbol{\omega})\mathrm{d}\boldsymbol{\omega}$$
$$\cdot \log \int p(y=c|\mathbf{x},\boldsymbol{\omega})q_\theta^*(\boldsymbol{\omega})\mathrm{d}\boldsymbol{\omega}$$
$$+ \mathbb{E}_{q_\theta^*(\boldsymbol{\omega})}\Big[\sum_c p(y=c|\mathbf{x},\boldsymbol{\omega})\log p(y=c|\mathbf{x},\boldsymbol{\omega})\Big]$$
$$\approx -\sum_c \Big(\frac{1}{T}\sum_t \widehat{p}_c^t\Big)\log\Big(\frac{1}{T}\sum_t \widehat{p}_c^t\Big)$$
$$+ \frac{1}{T}\sum_{c,t}\widehat{p}_c^t \log \widehat{p}_c^t =: \widehat{\mathbb{I}}[y,\boldsymbol{\omega}|\mathbf{x},\mathcal{D}_{\text{train}}]$$

defining our approximation, with $\widehat{p}_c^t$ the probability of input $\mathbf{x}$ with model parameters $\widehat{\boldsymbol{\omega}}_t \sim q_\theta^*(\boldsymbol{\omega})$ to take class c:

$$\widehat{\mathbf{p}}^t = [\widehat{p}_1^t, ..., \widehat{p}_C^t] = \text{softmax}(\mathbf{f}^{\widehat{\boldsymbol{\omega}}_t}(\mathbf{x})).$$

We then have

$$\widehat{\mathbb{I}}[y,\boldsymbol{\omega}|\mathbf{x},\mathcal{D}_{\text{train}}] \xrightarrow[T\to\infty]{} \mathbb{H}[y|\mathbf{x},q_\theta^*] - \mathbb{E}_{q_\theta^*(\boldsymbol{\omega})}\big[\mathbb{H}[y|\mathbf{x},\boldsymbol{\omega}]\big]$$
$$\approx \mathbb{I}[y,\boldsymbol{\omega}|\mathbf{x},\mathcal{D}_{\text{train}}],$$

resulting in a computationally tractable estimator approximating the BALD acquisition function. The other acquisition functions can be approximated similarly.

In the next section we will experiment with these acquisition functions and assess them empirically. These will be compared to the baseline acquisition function which uniformly acquires new data points from the pool set at random, and to various other techniques for active learning of image data and semi-supervised learning. This is followed by a real-world case study using cancer diagnosis.

5. Active Learning with Bayesian Convolutional Neural Networks

We study the proposed technique for active learning of image data. We compare the various acquisition functions relying on Bayesian CNN uncertainty with a simple image classification benchmark. We then study the importance of model uncertainty by evaluating the same acquisition functions with a deterministic CNN. This is followed by a comparison to a current technique for active learning with image data, which relies on SVMs. We follow with a comparison to the closest modern models to our active learning with image data – semi-supervised techniques with image data. These semi-supervised techniques have access to much more data (the unlabelled data) than our active learning models, yet we still perform in comparable terms to them. Finally, we demonstrate the proposed methodology with a real world application of skin cancer diagnosis from a small number of lesion images, relying on fine-tuning of a large CNN model.

5.1. Comparison of various acquisition functions

We next study all acquisition functions above with our Bayesian CNN trained on the MNIST dataset (LeCun & Cortes, 1998). All acquisition functions are assessed with the same model structure: convolution-relu-convolution-relu-max pooling-dropout-dense-relu-dropout-dense-softmax, with 32 convolution kernels, 4x4 kernel size, 2x2 pooling, dense layer with 128 units, and dropout probabilities 0.25 and 0.5 (following the example Keras MNIST CNN implementation (fchollet, 2015)).

All models are trained on the MNIST dataset with a (random but balanced) initial training set of 20 data points, and a validation set of 100 points on which we optimise the weight decay (this is a realistic validation set size, in comparison to the standard validation set size of 5K used in similar applications such as semi-supervised learning on MNIST). We further use the standard test set of 10K points, and the rest of the points are used as a pool set. The test error of each model and each acquisition function was assessed after each acquisition, using the dropout approximation at test time. To decide what data points to acquire though we used MC dropout following the derivations above. We repeated the acquisition process 100 times, each time acquiring the 10 points that maximised the acquisition function over the pool set. Each experiment was repeated three times and the results averaged (the standard deviation for the three repetitions is shown below)[2].

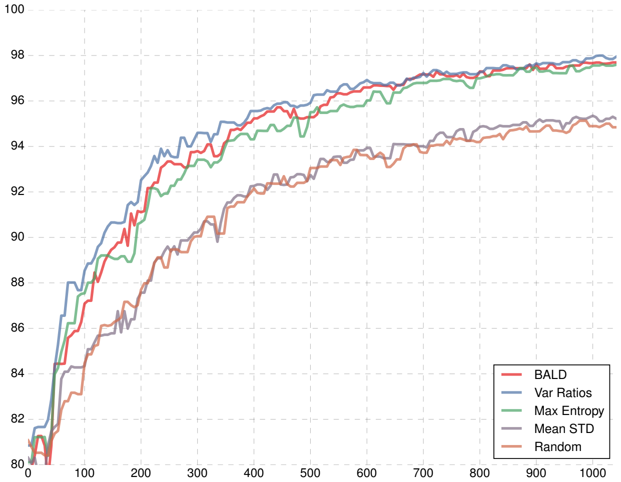
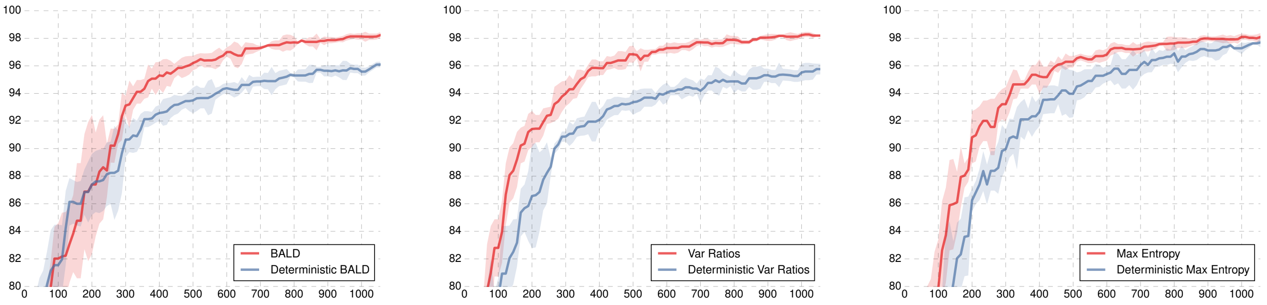


Figure 1. **MNIST test accuracy as a function of number of acquired images from the pool set** (up to 1000 images, using validation set size 100, and averaged over 3 repetitions). Four acquisition functions (*BALD*, *Variation Ratios*, *Max Entropy*, and *Mean STD*) are evaluated and compared to a *Random* acquisition function.

[2] The code for these experiments is available at



(a) BALD

(b) Var Ratios

(c) Max Entropy

Figure 2. Test accuracy as a function of number of acquired images for various acquisition functions, using both a **Bayesian CNN (red) and a deterministic CNN (blue)**.

We compared the acquisition functions BALD, Variation Ratios, Max Entropy, Mean STD, and the baseline Random. We found Random and Mean STD to under-perform compared to BALD, Variation Ratios, and Max Entropy (figure 1). The Variation Ratios acquisition function seems to obtain slightly better accuracy faster than BALD and Max Entropy. It is interesting that Mean STD seems to perform similarly to Random – which samples points at random from the pool set.

Lastly, in table 1 we give the number of acquisition steps needed to get to test errors of 5% and 10%. As can be seen, BALD, Variation Ratios, and Max Entropy attain a small test error with much fewer acquisitions than Mean STD and Random. This table demonstrates the importance of data efficiency – an expert using the Variation Ratios model for example would have to label less than half the number of images she would have had to label had she acquired new images at random.

% error	BALD	Var Ratios	Max Ent	Mean STD	Random
10%	145	120	165	230	255
5%	335	295	355	695	835

Table 1. Number of acquired images to get to model error of % on MNIST.

5.2. Importance of model uncertainty

We assess the importance of model uncertainty in our Bayesian CNN by evaluating three of the acquisition functions (BALD, Variation Ratios, and Max Entropy) with a deterministic CNN. Much like the Bayesian CNN, the deterministic CNN produces a probability vector which can be used with the acquisition functions of §4 (formally, by setting $q_\theta^*(\boldsymbol{\omega}) = \delta(\boldsymbol{\omega} - \theta)$ to be a point mass at the location of the model parameters θ). Such deterministic models can capture *aleatoric uncertainty* – the noise in the data – but cannot capture *epistemic uncertainty* – the uncertainty over the parameters of the CNN, which we try to minimise during active learning. The models in this experiment still use dropout, but for regularisation only (i.e. we do not perform MC dropout at test time).

A comparison of the Bayesian models to the deterministic models for the BALD, Variation Ratios, and Max Entropy acquisition functions is given in fig. 2. The Bayesian models, propagating uncertainty throughout the model, attain higher accuracy early on, and converge to a higher accuracy overall. This demonstrates that the uncertainty propagated throughout the Bayesian models has a significant effect on the models' measure of their confidence.

5.3. Comparison to current active learning techniques with image data

We next compare to a method in the sparse existing literature of active learning with image data, concentrating on (Zhu et al., 2003) which relies on a kernel method and further leverages the unlabelled images (which will be discussed in more detail in the next section). Zhu et al. (2003) evaluate an RBF kernel over the raw images to get a similarity graph which can be used to share information about the unlabelled data. Active learning is then performed by greedily selecting unlabelled images to be labelled, such that an estimate to the expected classification error is minimised. This will be referred to as *MBR*.

MBR was formulated for the binary classification case, hence we compared MBR to the acquisition functions BALD, Variation Ratios, Max Entropy, and Random on a binary classification task (two digits from the MNIST dataset). Classification accuracy is shown in fig. 3. Note that even a random acquisition function, when coupled with a CNN (a specialised model for image data) outperforms MBR which relies on an RBF kernel. We further experimented with a CNN version for MBR where we replaced the RBF kernel with a CNN. It is interesting to note that this did not give improved results.

http://mlg.eng.cam.ac.uk/yarin/publications.html#Gal2016Active.

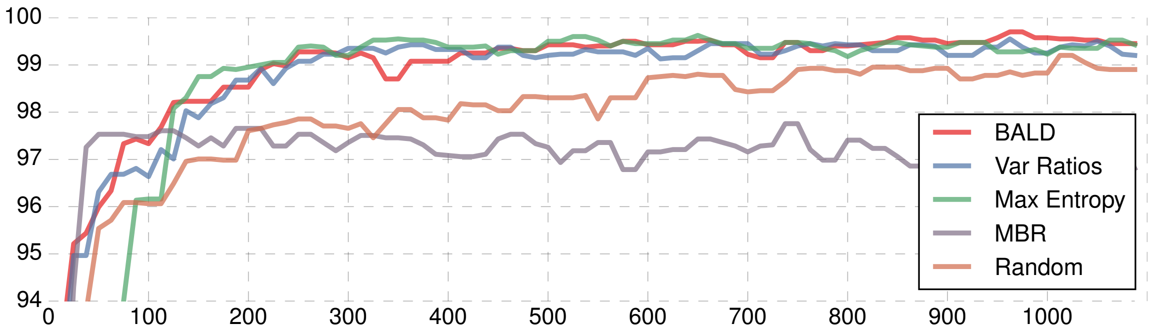


Figure 3. MNIST test accuracy (two digit classification) as a function of number acquired images, compared to a current technique for active learning of image data: MBR (Zhu et al., 2003).

5.4. Comparison to semi-supervised learning

We continue with a comparison to the closest models (in modern literature) to our active learning with image data: semi-supervised learning with image data. In *semi-supervised learning* a model is given a fixed set of labelled data, and a fixed set of unlabelled data. The model can use the unlabelled dataset to learn about the distribution of the inputs, in the hopes that this information will aid in learning the mapping to the outputs as well. Several semi-supervised models for image data have been suggested in recent years (Weston et al., 2012; Kingma et al., 2014; Rasmus et al., 2015), models which have set benchmarks on MNIST given a small number of *labelled* images (1000 random images). These models make further use of a (very) large unlabelled set of 49K images, and a large validation set of 5K-10K *labelled images* to tune model hyper-parameters and model structure (Rasmus et al., 2015). These models have access to much more data than our active learning models, but we still compare to them as they are the most relevant models in the field given the constraint of small amounts of *labelled* data. We compare to semi-supervised models which use a similar model structures to ours.

Test error for our active learning models with various acquisition functions (after the acquisition of 1000 training points), as well as the semi-supervised models, is given in table 2. In this experiment, to be comparable to the other techniques, we use a validation set of 5K points. Our model attains similar performance to that of the semi-supervised models (although note that we use a fairly small model compared to (Rasmus et al., 2015) for example). Rasmus et al. (2015)'s ladder network (full) attains error 0.84% with 1000 labelled images and 59,000 unlabelled images. However, Rasmus et al. (2015)'s Γ-model architecture is more directly comparable to ours. The Γ-model attains 1.53% error, compared to 1.64% error of our Var Ratio acquisition function which relies on no additional unlabelled data.

5.5. Cancer diagnosis from lesion image data

We finish by assessing the proposed technique with a real world test case. We experiment with melanoma (skin cancer) diagnosis from dermoscopic lesion images. In this task we are given image data of skin segments, of both malignant (cancerous) as well as benign (non-cancerous) lesions. Our task is to classify the images as malignant or benign (an example is shown in fig. 4). The data used is the ISIC Archive (Gutman et al., 2016). This dataset was collected in order to provide a "large public repository of expertly annotated high quality skin images" to provide clinical support in the identification of skin cancer, and to develop algorithms for skin cancer diagnosis. Specifically, we use the training data of the "ISBI 2016: Skin Lesion Analysis To-

Technique	Test error
Semi-supervised:	
Semi-sup. Embedding (Weston et al., 2012)	5.73%
Transductive SVM (Weston et al., 2012)	5.38%
MTC (Rifai et al., 2011)	3.64%
Pseudo-label (Lee, 2013)	3.46%
AtlasRBF (Pitelis et al., 2014)	3.68%
DGN (Kingma et al., 2014)	2.40%
Ladder Network (Γ-model) (Rasmus et al., 2015)	1.53%
Virtual Adversarial (Miyato et al., 2015)	1.32%
Active learning with various acquisitions:	
Random	4.66%
BALD	1.80%
Max Entropy	1.74%
Var Ratios	1.64%

Table 2. **Test error on MNIST with 1000 labelled training samples, compared to semi-supervised techniques.** Active learning has access to only the 1000 acquired images. Semi-supervised further has access to the remaining images with no labels. Following existing research we use a large validation set of size 5000.

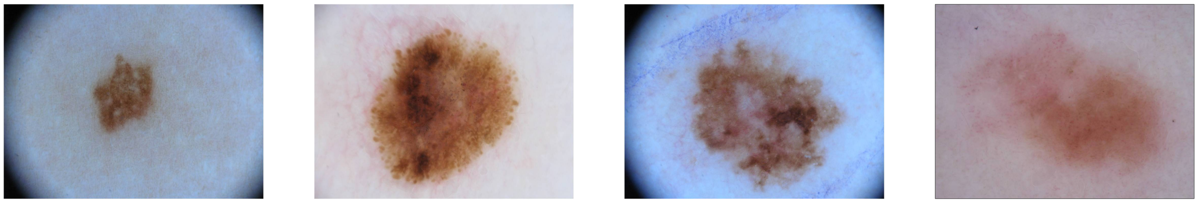


Figure 4. Skin cancer (melanoma) example lesions from the ISIC 2016 melanoma diagnosis dataset. The two lesions on the left are benign (non-cancerous), while the two lesions on the right are malignant (cancerous).

wards Melanoma Detection – Part 3B: Segmented Lesion Classification" task. The data contains 900 dermoscopic lesion images in JPEG format with EXIF tags removed. Malignancy diagnosis for these lesions was obtained from expert consensus and pathology report information. The data contains lesion segmentation as well, which we did not use.

For our model we replicate the model of (Agarwal et al., 2016). This model achieved second place in the "Part 3B: Segmented Lesion Classification" task, with its code open-sourced. The model relies on data augmentation of the positive examples (flipping the lesions vertically and horizontally), and fine-tunes the VGG16 CNN model (Simonyan & Zisserman, 2015) (i.e. optimises a pre-trained model with a small learning rate). The VGG16 model was pre-trained on ImageNet (Deng et al., 2009). The top layer of the model (1000 logits) was removed and replaced with a 2 dimensional output (for our classification task of malignant/benign). Preceding the last layer are two fully connected layers of size 4096, each one followed by a dropout layer with dropout probability 0.5. This architecture seems to provide good uncertainty estimates as observed before (Kendall et al., 2015; Gal & Ghahramani, 2016a).

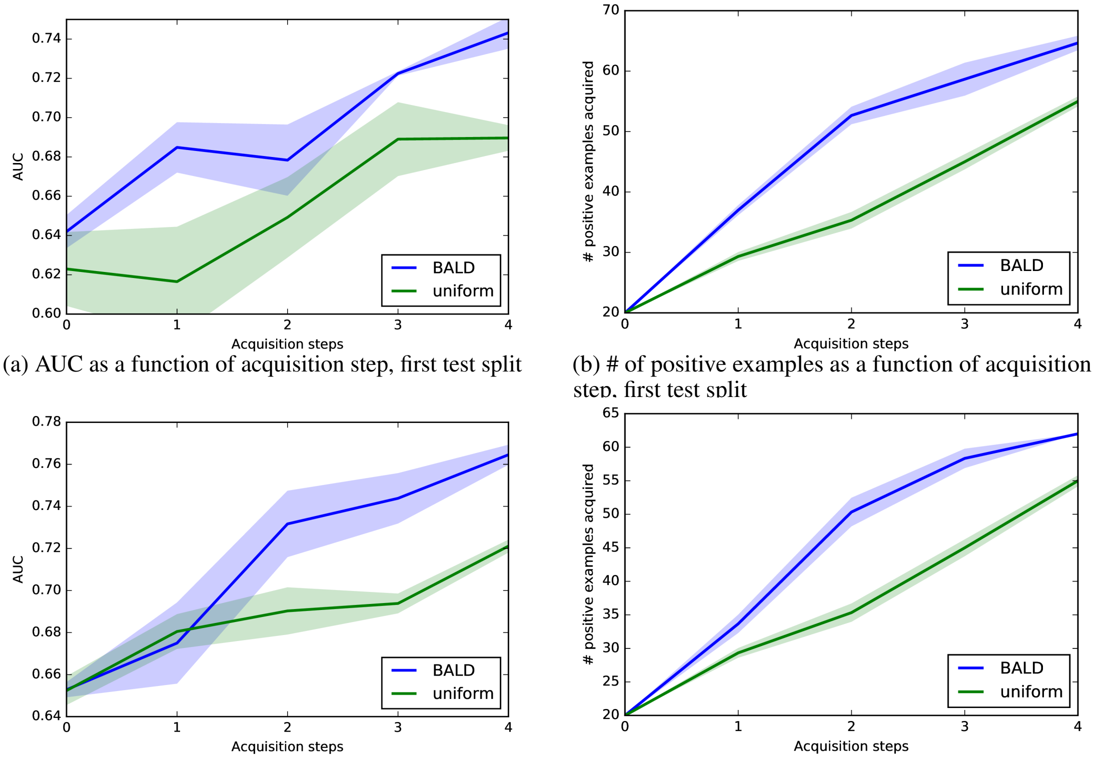
The data is unbalanced, containing 727 negative (benign) examples, and 173 positive (malignant) examples (20% positive examples). Since the data is so small, to assess model performance reliably we have to take a large balanced test set. We randomly partition the data, and set aside 100 negative and 100 positive examples. All our experiments are performed on two different random splits – since even a test set size of 200 gives very different accuracy with different random splits. Note that *on each such random split* we repeat our experiments three times and average the results with respect to the fixed test set.

We experiment with active learning by following the following procedure. We begin by creating an initial training set of 80 negative examples and 20 positive examples from our training data, as well as a pool set from the remaining data. With each experiment repetition (out of the three experiment repetitions w.r.t. the fixed test split) the pool is shuffled anew. The positive examples in the current training set are augmented following the original training procedure, and a model is trained on the augmented training set for 100 epochs until convergence. We use batch size 8 and weight decay set by $(1-p)l^2/N$, where N is the number of training points, $p = 0.5$ is the dropout probability, and the length-scale squared l^2 is set to 0.5. An acquisition function is then used to select the 100 most informative images from the pool set. These points are removed from the pool set and added to the (non-augmented) training set, where we use the original expert-provided labels for these points. The process is repeated until all pool points have been exhausted, where at each acquisition step we reset the model to its original pre-trained weights. This reset is done in order to avoid local optima, and to avoid confusing model performance improvement with an improvement resulting from simply using longer (cumulative) optimisation time.

After each acquisition the test performance of the model is logged using MC dropout with 20 samples. We further keep track of the number of positive examples acquired after each acquisition. Model performance is assessed using area-under-the-curve (AUC) as this seems to be the most informative of all metrics used by Gutman et al. (2016). We experimented with the *average precision* metric suggested by Gutman et al. (2016) as well, but managed to get results improving over the competition winner by simply predicting all points as "benign". This might be because of the data imbalance. AUC on the other hand takes into account all possible decision-thresholds possible to classify a malignant image.

We assessed two acquisition functions: a uniform baseline, and BALD. Even though Variation Ratios performs well on MNIST above, the function fails with the melanoma data since most malignant images are given only a slight higher probability of being malignant compared to the probability of benign images of being malignant. As a result all pool points are given identical Variation Ratios acquisition value.

Experiment results are given in fig. 5, where results are reported on both test splits (top and bottom), and where with each split the experiment is repeated three times and performance results are averaged on that fixed split. For each test split we report mean with standard error. AUC is reported for each split (left), and number of acquired positive examples is reported as well (right) for each acquisition step. BALD achieves better AUC faster than uniform, and acquires more positive examples at each acquisition step than uniform (i.e. BALD finds positive examples as informative and adds these to the training set, whereas uniform



(a) AUC as a function of acquisition step, first test split

(b) # of positive examples as a function of acquisition step, first test split

(c) AUC as a function of acquisition step, second test split

(d) # of positive examples as a function of acquisition step, second test split

Figure 5. AUC (left) as well as the number of acquired positive examples (right) for both the BALD acquisition function as well as uniform acquisition function, on **ISIC 2016 melanoma diagnosis dataset**. Two random test splits are assessed (top and bottom), and on each test set the experiment was repeated three times with different random seeds (shown mean with standard error).

simply selects positive examples from the pool set based on their frequency).

Note how AUC range varies wildly between the two different test splits, but how AUC is similar for both acquisition functions on each fixed test set before the initial acquisition (when both uniform and BALD models are trained on the same initial training set). This demonstrates the difficulties with handling of small data: each test split gives radically different results, and in this case even though each acquisition function experiment has a relatively small standard error, averaging the AUC of the acquisition functions over the different test splits would artificially increase the standard error. Lastly, it is interesting to experiment with a model trained over the entire pool set, i.e. with the settings of the second place winner in the ISIC2016 task. For the first test split this model attains AUC 0.71 ± 0.003, whereas with the second test split it attains AUC 0.75 ± 0.01. For both test splits this AUC is worse than BALD's converged AUC after 4 acquisition steps. This might be because BALD avoided selecting noisy points – near-by images for which there exist multiple noisy labels of different classes. Such points have large aleatoric uncertainty – uncertainty which cannot be explained away – rather than large epistemic uncertainty – the uncertainty which BALD captures *in order* to explain it away, i.e. reduce it.

6. Future Research

We presented a new approach for active learning of image data, relying on recent advances at the intersection of Bayesian modelling and deep learning, and demonstrated a real-world application in medical diagnosis. We assessed the performance of the techniques by resetting the models after each acquisition, and training them again to convergence. This was done to isolate the effects of our acquisition functions, which came at a cost of prolonged training times (20 hours for each melanoma experiment for example). We showed that even with this long running time, our technique still reduces required expert labels, thus reduces costs for such a system. This running time can be reduced further by not resetting the system – with the potential price of falling into local optima. We leave this problem for future research.

References

Agarwal, Mohit, Damaraju, Nandita, and Chaieb, Sahbi. Dl8803. https://github.com/NanditaDamaraju/DL8803, 2016.

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

Cortes, Corinna and Vapnik, Vladimir. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. IEEE, 2009.

fchollet. Keras. https://github.com/fchollet/keras, 2015.

Freeman, Linton G. Elementary applied statistics, 1965.

Gal, Yarin. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

Gal, Yarin and Ghahramani, Zoubin. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *ICLR workshop track*, 2016a.

Gal, Yarin and Ghahramani, Zoubin. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *ICML*, 2016b.

Gutman, David, Codella, Noel CF, Celebi, Emre, Helba, Brian, Marchetti, Michael, Mishra, Nabin, and Halpern, Allan. Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC). *arXiv preprint arXiv:1605.01397*, 2016.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

Hernandez-Lobato, Jose Miguel and Adams, Ryan. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 1861–1869, 2015.

Hinton, Geoffrey E, Srivastava, Nitish, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Holub, Alex, Perona, Pietro, and Burl, Michael C. Entropy-based active learning for object recognition. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pp. 1–8. IEEE, 2008.

Houlsby, Neil, Huszár, Ferenc, Ghahramani, Zoubin, and Lengyel, Máté. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Joshi, Ajay J, Porikli, Fatih, and Papanikolopoulos, Nikolaos. Multi-class active learning for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2372–2379. IEEE, 2009.

Jozefowicz, Rafal, Vinyals, Oriol, Schuster, Mike, Shazeer, Noam, and Wu, Yonghui. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

Kalchbrenner, Nal and Blunsom, Phil. Recurrent continuous translation models. In *EMNLP*, 2013.

Kampffmeyer, Michael, Salberg, Arnt-Borre, and Jenssen, Robert. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

Kendall, Alex, Badrinarayanan, Vijay, and Cipolla, Roberto. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

Kingma, Diederik P, Mohamed, Shakir, Rezende, Danilo Jimenez, and Welling, Max. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pp. 3581–3589, 2014.

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

LeCun, Yann and Cortes, Corinna. The MNIST database of handwritten digits, 1998.

LeCun, Yann, Boser, Bernhard, Denker, John S, Henderson, Donnie, Howard, Richard E, Hubbard, Wayne, and Jackel, Lawrence D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

Lee, Dong-Hyun. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning*, 2013.

Li, Xin and Guo, Yuhong. Adaptive active learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 859–866, 2013.

Marcus, Daniel S, Fotenos, Anthony F, Csernansky, John G, Morris, John C, and Buckner, Randy L. Open access series of imaging studies: longitudinal mri data in nondemented and demented older adults. *Journal of cognitive neuroscience*, 22(12):2677–2684, 2010.

Miyato, Takeru, Maeda, Shin-ichi, Koyama, Masanori, Nakae, Ken, and Ishii, Shin. Distributional smoothing by virtual adversarial examples. *arXiv preprint arXiv:1507.00677*, 2015.

Pitelis, Nikolaos, Russell, Chris, and Agapito, Lourdes. Semi-supervised learning using an unsupervised atlas. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 565–580. Springer, 2014.

Rasmus, Antti, Berglund, Mathias, Honkala, Mikko, Valpola, Harri, and Raiko, Tapani. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pp. 3546–3554, 2015.

Rifai, Salah, Dauphin, Yann N, Vincent, Pascal, Bengio, Yoshua, and Muller, Xavier. The manifold tangent classifier. In *Advances in Neural Information Processing Systems*, pp. 2294–2302, 2011.

Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

Shannon, Claude Elwood. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 2014.

Sundermeyer, Martin, Schlüter, Ralf, and Ney, Hermann. LSTM neural networks for language modeling. In *INTERSPEECH*, 2012.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc VV. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

Tong, Simon. *Active Learning: Theory and Applications*. PhD thesis, 2001. AAI3028187.

Weston, Jason, Ratle, Frédéric, Mobahi, Hossein, and Collobert, Ronan. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pp. 639–655. Springer, 2012.

Zhu, X, Lafferty, J, and Ghahramani, Z. Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data*, pp. 58–65. ICML, 2003.