Deep Bayesian Active Learning with Image Data

Yarin Gal **Riashat Islam** **Zoubin Ghahramani**
University of Cambridge
{yg279,ri258,zg201}@cam.ac.uk

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, we have 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. Second, many acquisition functions rely on model uncertainty. In deep learning on the other hand 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. Taking advantage of specialised models such as Bayesian convolutional neural networks, we demonstrate our active learning techniques with image data, obtaining significant improvement on existing active learning approaches.

1 Introduction

A big challenge in many applications is obtaining labelled data. This can be a long and laborious process, which often makes the development of an automated system 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 applicable to a wider class of problems. Such frameworks for learning are referred to as *active learning* [1] (also known as "experiment design" in the statistics literature), and have been used successfully in fields such as medical diagnosis, microbiology, and manufacturing [2]. 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 what 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.

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 [2]. This data appears often in image form, with a physician classifying MRI scans to diagnose Alzheimer's for example [3], 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 [2, 4], with only a handful of exceptions [5–7] relying on kernel or graph-based approaches.

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 [8–11]. And with the

Workshop on Bayesian Deep Learning, NIPS 2016, Barcelona, Spain.

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) [12, 13] 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 the task of object recognition (given enough training data) [14]. New techniques such as dropout [15, 16] are used extensively to regularise these huge models, which often contain millions of parameters [17]. 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 handle small amounts of data. Recent advances in deep learning, on the other hand, are notorious for their dependence on large amounts of data [9]. Second, many acquisition functions rely on model uncertainty. In deep learning on the other hand 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 [5, 18, 6, 7]. Taking advantage of specialised models such as Bayesian convolutional neural networks (BCNNs) [19, 20], 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 much 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 [21] or 1.53% test error of the Ladder Network Γ-model [22], both semi-supervised learning techniques which additionally use the entire unlabelled training set.

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 [2], Joshi et al. [7] used "margin-based uncertainty" and extracted probabilistic outputs from support vector machines (SVM) [23]. They used linear, polynomial, and Radial Basis Function (RBF) kernels on the raw images, picking the kernel that gave best classification accuracy. Contrary to SVM approaches, Li and Guo [18] used Gaussian processes (GPs) with RBF kernels to get model uncertainty. However Li and Guo [18] fed low dimensional features (such as SIFT features) to their RBF kernel. Lastly, making use of unlabelled data as well, Zhu et al. [5] 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 literate includes semi-supervised learning of image data [24, 21, 22]. 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.

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 [5, 18, 7] 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) [12, 13]. 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 [9]. To perform active learning with image data we make use of the Bayesian equivalent of CNNs, proposed in [19][1]. These Bayesian CNNs are

[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.

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 [15, 16], originally used to regularise these models. As shown in [20, 25] 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 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 [25].

Bayesian CNNs work well with small amounts of data [19], and possess uncertainty information that can be used with existing acquisition functions [25]. Such acquisition functions for the case of classification are discussed next.

4 Acquisition Functions and their Approximations

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 would often use the predictive variance for our acquisition function. 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 for us:

1. Choose pool points that maximise the predictive entropy (*Max Entropy*, [26])

$$\mathbb{H}[y|\mathbf{x}, \mathcal{D}_{\text{train}}] := -\sum_c p(y = c|\mathbf{x}, \mathcal{D}_{\text{train}}) \log p(y = c|\mathbf{x}, \mathcal{D}_{\text{train}}).$$

2. Maximise the mutual information between predictions and model posterior (*BALD*, [27])

$$\mathbb{I}[y, \boldsymbol{\omega}|\mathbf{x}, \mathcal{D}_{\text{train}}] = \mathbb{H}[y|\mathbf{x}, \mathcal{D}_{\text{train}}] - \mathbb{E}_{p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})}\big[\mathbb{H}[y|\mathbf{x}, \boldsymbol{\omega}]\big]$$

with $\boldsymbol{\omega}$ the model parameters. Points that maximise this acquisition function are points on which the model is uncertain on average, but there exist model parameters that produce erroneous 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 *Variation Ratios* [28]

$$\text{variation-ratio}[\mathbf{x}] := 1 - \frac{f_{\mathbf{x}}}{T}$$

with $f_{\mathbf{x}} = \sum_t \mathbb{1}[y^t = c^*]$ and c^* being the mode of $\{y^t\}$ (a set of samples from the predictive distribution at input $\mathbf{x}$).

4. Maximise *Mean STD* [29, 30]

$$\sigma(\mathbf{x}) = \frac{1}{C} \sum_c \sqrt{\mathbb{E}_{q(\boldsymbol{\omega})}[p(y = c|\mathbf{x}, \boldsymbol{\omega})^2] - \mathbb{E}_{q(\boldsymbol{\omega})}[p(y = c|\mathbf{x}, \boldsymbol{\omega})]^2}$$

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): $g(\mathbf{x}) = \frac{1}{N}$ with N pool points.

These acquisition functions and their properties are discussed in more detail in [25, 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:

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

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 $p(y = c|\mathbf{x}, \mathcal{D}_{\text{train}}) = \int p(y = c|\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}}) \mathrm{d}\boldsymbol{\omega}$:

$$\begin{aligned}
\mathbb{I}[y, \boldsymbol{\omega}|\mathbf{x}, \mathcal{D}_{\text{train}}] = &-\sum_c \int p(y = c|\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}}) \mathrm{d}\boldsymbol{\omega} \log \int p(y = c|\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}}) \mathrm{d}\boldsymbol{\omega} \\
&+ \mathbb{E}_{p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})}\left[\sum_c p(y = c|\mathbf{x}, \boldsymbol{\omega}) \log p(y = c|\mathbf{x}, \boldsymbol{\omega})\right].
\end{aligned}$$

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:

$$\begin{aligned}
&\approx -\sum_c \int p(y = c|\mathbf{x}, \boldsymbol{\omega}) q^*_\theta(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \log \int p(y = c|\mathbf{x}, \boldsymbol{\omega}) q^*_\theta(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \\
&\quad + \mathbb{E}_{q^*_\theta(\boldsymbol{\omega})}\left[\sum_c p(y = c|\mathbf{x}, \boldsymbol{\omega}) \log p(y = c|\mathbf{x}, \boldsymbol{\omega})\right] \\
&\approx -\sum_c \left(\frac{1}{T} \sum_t \widehat{p}^t_c\right) \log \left(\frac{1}{T} \sum_t \widehat{p}^t_c\right) + \frac{1}{T} \sum_{c,t} \widehat{p}^t_c \log \widehat{p}^t_c := \widehat{\mathbb{I}}[y, \boldsymbol{\omega}|\mathbf{x}, \mathcal{D}_{\text{train}}]
\end{aligned}$$

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

$$\widehat{\mathbf{p}}^t = [\widehat{p}^t_1, ..., \widehat{p}^t_C] = \text{softmax}(\mathbf{f}^{\widehat{\boldsymbol{\omega}}_t}(\mathbf{x})).$$

We then have

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

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 a 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.

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 finish 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 than our active learning models, yet we still perform in comparable terms to them.

5.1 Comparison of various acquisition functions

We next study all acquisition functions above with our Bayesian CNN trained on the MNIST dataset [31]. 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 [32]).

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 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

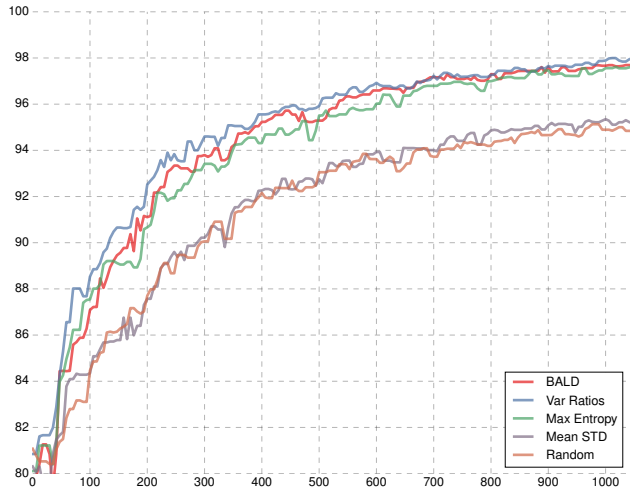


Figure 1: **MNIST test accuracy as a function of number of acquired images from the pool set** (up to 1000 images, using validation set size 100, and averaged over 3 repetitions). Four acquisition functions (*BALD*, *Variation Ratios*, *Max Entropy*, and *Mean STD*) are evaluated and compared to a *Random*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

Technique	Test error
Semi-supervised:	
Semi-sup. Embedding [24]	5.73%
Transductive SVM [24]	5.38%
MTC [33]	3.64%
Pseudo-label [34]	3.46%
AtlasRBF [35]	3.68%
DGN [21]	2.40%
Virtual Adversarial [36]	1.32%
Ladder Network (Γ-model) [22]	1.53%
Ladder Network (full) [22]	0.84%
Active learning with various acquisitions:	
Random	4.66%
BALD	1.80%
Max Entropy	1.74%
Var Ratios	1.64%

Figure 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 val. set of 5000.

acquisition function over the pool set. Each experiment was repeated 3 times and the results averaged (the standard deviation for the 3 repetitions is shown in fig. 3 below)[2].

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.

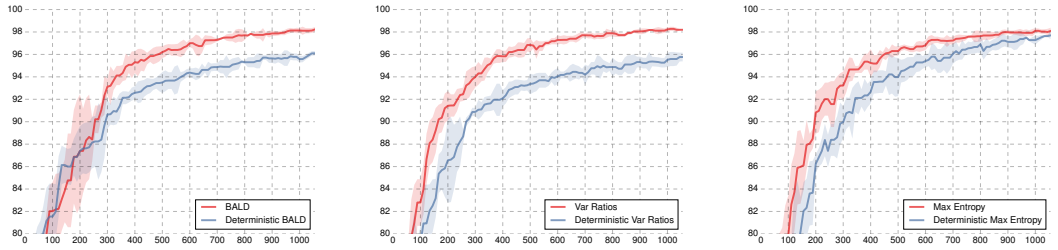
% error	BALD	Var Ratios	Max Entropy	Mean STD	Random
10%	145	120	165	230	255
5%	335	295	355	695	835

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. 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. 3. 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.



(a) BALD (b) Var Ratios (c) Max Entropy

Figure 3: 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)**.

[2]The code for these experiments is available at
`https://github.com/Riashat/Active-Learning-Bayesian-Convolutional-Neural-Networks/tree/master/ConvNets/FINAL_Averaged_Experiments/Final_Experiments_Run`.

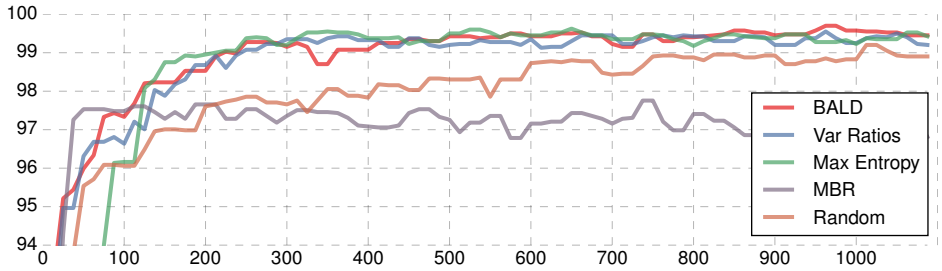


Figure 4: MNIST test accuracy (two digit classification) as a function of number acquired images, compared to a current technique for active learning of image data: MBR [5].

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 [5] 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. [5] 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. 4. 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.

5.4 Comparison to semi-supervised learning

We finish 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 [24, 21, 22], 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 [22]. 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.

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 [22] for example). Rasmus et al. [22]'s ladder network (full) attains error 0.84% with 1000 labelled images and 59,000 unlabelled images. However, [22]'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.

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. This approach would hopefully lay the way to a variety of new applications in medical diagnosis, microbiology, and manufacturing. Future research

includes the extension of the ideas above to more complex models, able to represent better uncertainty estimates, and capture more complex data.

References

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

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

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

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

[5] X Zhu, J Lafferty, and Z Ghahramani. 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*, pages 58–65. ICML, 2003.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[21] 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*, pages 3581–3589, 2014.

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

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

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

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

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

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

[28] Linton G Freeman. Elementary applied statistics, 1965.

[29] Michael Kampffmeyer, Arnt-Borre Salberg, and Robert Jenssen. 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.

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

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

[32] fchollet. Keras. `https://github.com/fchollet/keras`, 2015.

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

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

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

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