Deep Bayesian Active Learning with Hybrid Query Strategies: Final Report

G073 (s2037105, s2342643, s1911076)

Abstract

Active learning is an effective technique for reducing expert supervision with annotating costs by selectively querying unannotated data points. It has been shown powerful in improving learning efficiency for neural networks, especially Bayesian neural networks, an architecture suitable for small datasets. However, traditional active learning usually query data points based solely on uncertainty or diversity, which can cause myopic decision boundaries or low query efficiency. To test whether query strategies considering both uncertainty and diversity can further accelerate learning, we develop 54 hybrid query strategies (3 uncertainty metrics * 3 diversity metrics * 6 combination methods), and compare them with pure query strategies on the MINST and CIFAR-10 datasets in a Bayesian convolutional neural network framework. We find query strategy performances are adjusted by the task difficulty and number of queried points. Hybrid query strategies shine in the difficult task with limited queried points. Their advantages decrease but still exist with more queried points.

1. Introduction

Active Learning (AL) is a machine learning method that uses human-in-the-loop systems to ask a human to annotate data points and use them to train models. It aims to achieve acceptable performances with less supervision by proper query strategies. The query strategies have been extensively studied in various model structures, including support vector machines (Joshi et al., 2009) and Gaussian processes (GPs) (Li & Guo, 2013) in classical AL, and random forest (Zeni et al., 2019) and incremental GPs (Bontempelli et al., 2020) in skeptical Learning as an extension of AL.

A recent promising direction in AL is to combine it with neural networks (Ren et al., 2021), which are known for their strong representation ability. On the one hand, neural networks are notorious for the dependence of big datasets, but manually annotating a large amount of points can be costly. AL is designed to solve this problem. On the other hand, the scalability to high dimensional data like images is a challenge for traditional AL methods (Tong, 2001), while it can be mitigated by neural networks.

There have been some successful attempts. Ash et al. (2019) apply neural networks using batch[1] active learning by diverse gradient embeddings in image classification tasks and achieve better accuracy consistently than other architectures. In succession, Ash et al. (2021) use a fisher-based active selection objective in a neural network, the model outperform the previous state of the art models on both classification and regression problems. Inspired by it, Teso & Vergari (2022) apply probabilistic circuits to make deep AL models reliable. Among current neural network structures, Bayesian neural networks (BNNs) (MacKay, 1992) are a strong candidate. Classical neural networks can overfit quickly on small datasets, but BNNs are robust to overfitting. In the meantime, BNNs encode uncertainty in parameters (epistemic uncertainty) explicitly, thus can offer a better estimate of prediction uncertainty that helps to query new points for AL. Actually, Gal et al. (2017) have built a Bayesian convolutional neural network (BCNN) with AL for image classification. They find the model perform better than deterministic CNNs over different query strategies. And it is comparable to two semi-supervised neural networks but using much less data. However, their, and most previous deep AL studies (e.g., Sener & Savarese (2017); Ducoffe & Precioso (2018)) have a limitation in query strategies. Their query strategies only consider uncertainty or diversity of unannotated data points, while those strategies can meet problems in specific situations.

A burgeoning direction in deep AL is to consider information from both sides (Zhdanov, 2019; Yin et al., 2017). However, to our best knowledge, there have been no studies systematically comparing pure and hybrid query strategies in deep AL, not to mention BNNs [2]. To fill this gap, this study aims to compare pure and hybrid query strategies in a BCNN framework. Our main contributions are as following:

1.We propose two computationally efficient diversity metrics, and develop 54 hybrid query strategies using them and other existing uncertainty and diversity metrics. These strategies are compared with the pure query strategies on the MNIST and CIFAR-10 datasets.

2.We show that the performances of query strategies are adjusted by the task difficulty and number of queried points.

[1]Unless specified, "batch" means the group of unannotated points being selected in one query and "batch size" means the size of that group in this report.

[2]Some studies, like Sener & Savarese (2017) use the models of Gal et al. (2017) as a baseline when comparing different query strategies. But their main model and other baselines are not BNNs. This is somewhat a misunderstanding since BNNs only provide a better estimate of uncertainty.

When the task is simple (MNIST), uncertainty-based query strategies are enough to accelerate learning. When the task is difficult (CIFAR-10) with limited queried points, diversity-based query strategies are more efficient among pure query strategies. This pattern is reversed when more points are queried. Hybrid query strategies can further accelerate learning than pure query strategies in the difficult task. Their advantages decrease but still exist when more points are queried.

2. Datasets and task

All models are trained on the MNIST dataset (Yann et al., 1998) first to reproduce the results of Gal et al. (2017), and to preliminarily compare the performances of different query strategies. A further experiment on CIFAR-10 (Krizhevsky et al., 2009) dataset is then applied to test whether the performance is adjusted by task difficulty.

MNIST is a popular dataset of handwritten digits that is commonly used in machine learning and computer vision research. It consists of 70,000 one-channel images of size 28x28 pixels, each representing a handwritten digit from 0 to 9. Each image is transformed to a tensor and normalized using mean and standard deviation computed over the dataset before training. The mean and standard deviation are 0.1307 and 0.3081, respectively. Those images are split into a training set of 60,000 images and a test set of 10,000 images.

CIFAR-10 dataset consists of 60,000 three-channel images of size 32x32 pixels, each representing one of the ten object classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Similarly, each image is transformed to a tensor and normalized before training. The computed mean and the standard deviation are (0.4914, 0.4822, 0.4465), (0.2023, 0.1994, 0.2010) for three channels, respectively. The images are split into a training set of 50,000 images and a test set of 10,000 images. This dataset is more challenging for image classification compared to the MNIST dataset due to its complexity and variability.

The main task of this study is to compare the performances of BCNNs with different query strategies on image classification tasks. The evaluation metric is the final accuracy on the test set. A higher final accuracy indicates a stronger query strategy. Accuracy as a function of query times is also used to analyse whether the performances of query strategies are stable during the learning process.

3. Methodology

3.1. Bayesian convolutional neural networks

As mentioned above, classical CNNs face the problem of quick overfitting on small datasets, BCNNs address the problem by putting priors over parameters, or more specifically, weights. Inferring the posterior of BCNNs is hard, so one would usually use a tractable variational distribution like Gaussian to approximate it (Hinton & Van Camp, 1993). This method is still computationally costly since considerable extra parameters are added to the neural networks. Fortunately, Gal & Ghahramani (2015b) show a common technique, dropout (Srivastava et al., 2014), is equivalent to sampling from a special variational distribution (a mixture of Bernoulli mixture and Gaussian mixture) in BNNs. So minimizing the loss function for a classical CNN with dropout is equivalent to minimizing the Kullback-Leibler divergence between the true posterior and the variational distribution in a BCNN. And keeping dropout at the prediction stage is equivalent to using sampled weights from the posterior to make predictions. Based on this, training a BCNN can be as quick as training a CNN, and prediction uncertainty can be calculated from several forward propagations with dropout on the test instance. Gal & Ghahramani (2015a) built a CNN and compare its performance with and without dropout on the test set (the former is equivalent to a BCNN and the final prediction is made by averaging several forward propagations) in a image classification task, and find the former achieves a significantly higher accuracy.

3.2. Query strategies

The core property of AL is it can select the most valuable data point (or data batch in neural networks) from the unannotated data pool. Thus, the choice of query strategies [3] can directly influence AL's performance. Query strategies in AL can be roughly divided into three branches: Uncertainty-based, diversity-based and their hybrid versions.

3.2.1. Uncertainty-based query strategies

Uncertainty-based query strategies will rank the unannotated data points based on uncertainty and choose the top K points with the highest uncertainty. Uncertainty can be categorized into aleatoric and epistemic uncertainty (Nguyen et al., 2019). The former is caused by the inherent noise inside data (e.g., two data points with the same features can belong to different classes). The latter is caused by the learning from limited data (e.g., the estimates of parameters will fluctuate less with more data). Deterministic neural networks can only capture the former but BNNs can capture both where the latter is reflected on the variance in sampled weights from the posterior. In this study, we consider three uncertainty metrics: entropy, Bayesian active learning by disagreements (BALD) and variation ratio.

Entropy For a classification task, entropy (Shannon, 1948) is a natural value to measure uncertainty since the results follow a categorical distribution. The predictive entropy of an unannotated data point $\boldsymbol{x}$[4] is

$$E(\boldsymbol{x}) = -\sum_{c} p(\boldsymbol{x} \in c|\boldsymbol{x}, M) log p(\boldsymbol{x} \in c|\boldsymbol{x}, M)$$

Here $p(\boldsymbol{x} \in c|\boldsymbol{x}, M)$ is the predicted probability that the point belongs to class c under model M. In a BCNN, it is

[3] they are also called acquisition functions in some literature like Gal et al. (2017).

[4] Strictly speaking, this is the feature or embedding of feature of that point. But we do not distinguish them for simplicity.

calculated by $\sum_{t=1}^{T} p_t(\boldsymbol{x} \in c|\boldsymbol{x}, M)$, where T is the number of forward propagations to be conducted and $p_t(\boldsymbol{x} \in c|\boldsymbol{x}, M)$ is the prediction of the t^{th} forward propagation. Each propagation should generate a different prediction due to dropout.To scale its range to [0, 1], which is important when combining metrics, we divide it by $log(N)$ where N is the number of possible classes.

BALD Entropy is usually not very effective in traditional neural networks with AL because they can be overconfident on small datasets. This problem is partly mitigated by BCNNs. But a more proper uncertainty metric for BCNNs is BALD (Houlsby et al., 2011):

$$B(\boldsymbol{x}) = E(\boldsymbol{x}) - \sum_{t=1}^{T} E_t(\boldsymbol{x})$$

Here $E_t(\boldsymbol{x})$ is the entropy calculated from the t^{th} forward propagation. If the propagation results vary much, the first term above will be high while the second term will be low since the model changes the predicted class of $\boldsymbol{x}$ from time to time. So BALD mainly considers epistemic uncertainty. We also divide it by $log(N)$ to scale its range.

Variation Ratio Similar to entropy, variation ratio (Freeman, 1965) measures the lack of confidence. It is calculated by

$$V(\boldsymbol{x}) = 1 - \max_{c} p(\boldsymbol{x} \in c|\boldsymbol{x}, M)$$

Since BCNNs themselves somewhat realize ensemble modeling, variation ratio serves as a commitee-based metric (Seung et al., 1992). Specially, $p(\boldsymbol{x} \in c|\boldsymbol{x}, M)$ here is not the mean predicted probability that $\boldsymbol{x}$ belongs to c in practice. Instead, each forward propagation will give a predicted class for $\boldsymbol{x}$. And $p(\boldsymbol{x} \in c|\boldsymbol{x}, M)$ is the ratio that the prediction is c among the forward propagations. The range of variation ratio is naturally [0, 1].

However, only considering uncertainty in prediction is not enough. Such strategies can be fooled by adversarial examples (Ducoffe & Precioso, 2018). That is, they can only focus on points around the current decision boundary but ignore the true distribution of data. Figure 1 gives an example. It illustrates a binary classification task where the true decision boundary is a triangle. Unfortunately, the initial training set only contains points around the left edge of the triangle. Uncertainty-based query strategies will continuously query points around that edge (suppose most points are distributed within and around the triangle). Since AL usually does not query many points, it is very likely the model will finally be confident that the true decision boundary is the edge. In other words, uncertainty-based query strategies can stuck in a myopic decision boundary when the task is difficult (i.e., a complex true decision boundary), while diversity-based query strategies can solve this problem.

3.2.2. Diversity-based query strategies

The definition of diversity itself is "diverse". In some literature, it refers to how representative the selected points

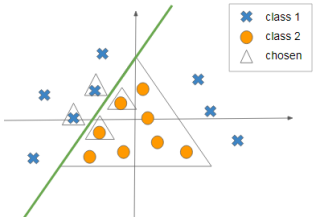

Figure 1. An example of a myopic decision boundary.

are of the whole unannotated pool. Corresponding query strategies usually tend to construct a core set as a surrogate for the whole pool (Sener & Savarese, 2017; Geifman & El-Yaniv, 2017). To some extent, uniformly choosing points from the pool is also a diversity-based query strategy since it implicitly matches the distribution. Another definition is the diversity among the selected points. Corresponding query strategies will select points far away from each other (Yin et al., 2017). That is, it prefers points covering the entire embedding space (the inputs in image classification tasks are usually high-dimensional. Calculating the diversity in the original space is inefficient and somewhat meaningless. Fortunately, CNNs provide a natural way to reduce dimensions and we call the low-dimensional space embedding space since its dimensions usually represent task-relevant features) but ignoring the actual distribution. The third definition is the similarity between the annotated and unannotated points. Corresponding query strategies will select points far from the annotated points (Gissin & Shalev-Shwartz, 2019; Shui et al., 2019). We follows this definition since the relevant metrics are computationally efficient. In AL tasks, annotated points are typically much less than the unannotated pool. Diversity based on the former two definitions usually require numerous traversal over the pool or sequential selection. But diversity based on the third definition mainly traverses over the annotated points and can be calculated in parallel. In this study, we consider three diversity metrics: Discriminator score, posterior variance and minimum distance. The latter two are modified from metrics used by previous studies.

Discriminator score The core idea of discriminator score (Gissin & Shalev-Shwartz, 2019) is to introduce a discriminator. It is trained on the pool and the annotated points to distinguish whether a point is annotated or not. Then it can generate the predicted probability of being unannotated for each points in the pool (i.e., the unannotated points are used in both training and testing). That is:

$$D(\boldsymbol{x}) = p(\boldsymbol{x} \in unannotated|\boldsymbol{x}, M_1)$$

Here M_1 is the discriminator model. It is usually a smaller neural network since the image has been projected to the embedding space. To solve the problem of imbalanced samples, we re-sample the annotated points until their number is equal to the pool size. The range of discriminator score is naturally [0, 1].

Posterior variance Posterior variance is a metric based on GPs. GPs define beliefs over function values (Rasmussen et al., 2006). Specifically, suppose there is function map-

ping each point in a given space to a real value (the value itself is unimportant here). GPs assume the value for any point follows the same Gaussian distribution if no values having been observed. However, if values of some points have been observed, we can infer the posterior distribution of value for any other point. It is still a Gaussian distribution, but its variance will decrease. The extent of decrease is higher when the similarity (i.e., an "overall distance") between the point interested and the points whose values are observed is larger. Li & Guo (2013) use this decrease of variance as a diversity metric. However, they are interested in this decrease for a given point when the values of all other points in the pool are observed, and decide to choose the point whose decrease is the largest. It is follows the first definition discussed above and can be computationally inefficient when the pool size is large. Instead, we define the diversity of an unannotated point as the posterior variance of its corresponding value after observing the values of all annotated points. To calculate it, we first choose the radio basis kernel function to measure the similarity between any two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2d})$$

where exp is the exponential function, $||\boldsymbol{x}_i - \boldsymbol{x}_j||^2$ is the square of Euclidean distance between the two points and d is the embedding dimension. $2d$ is chosen heuristically since our embedding data is normalized before calculation. The expectation of the square of Euclidean distance between two arbitrarily selected points is $2d$. Then the posterior variance is calculated using

$$P(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - k_{\boldsymbol{x},\boldsymbol{Y}} K(\boldsymbol{Y})^{-1} k_{\boldsymbol{x},\boldsymbol{Y}}^T$$

where $k(\boldsymbol{x}, \boldsymbol{x})$ is actually 1, T and $^{-1}$ are the transpose and inverse symbols. $\boldsymbol{Y}$ indicates the set of annotated points and we use $\boldsymbol{y}_i$ to denote the i^{th} annotated point. If there are n annotated points, $k_{\boldsymbol{x},\boldsymbol{Y}}$ is a n-dimensional vector whose i^{th} element is $k(\boldsymbol{x}, \boldsymbol{y}_i)$ and $K(\boldsymbol{Y})$ is a $n * n$ matrix whose (i, j) element is $k(\boldsymbol{y}_i, \boldsymbol{y}_j)$. The range of posterior variance is [0, 1].

Minimum distance Minimum distance is a heuristic method to measure the distance between one point and a group of points. It is inspired by the core set construction work of Sener & Savarese (2017). In their study, when some cluster centers are given, the next cluster center is the point whose minimum distance to the existing centers is maximum. Because they search centers in the pool and use them as the core set, this method is computationally inefficient and can not be conducted in parallel. In our modified version, the "centers" are all annotated points and the minimum distance is calculated by

$$M(x) = \min_i ||\boldsymbol{x} - \boldsymbol{y}_i||$$

where $||\boldsymbol{x} - \boldsymbol{y}_i||$ is the Euclidean distance between $\boldsymbol{x}$ and the i^{th} annotated point. To scale the range of minimum distance, we divide it by the maximum minimum distance in the pool.

However, the limitations of diversity-based query strategies are also obvious. First, the calculation of diversity metrics are computationally costly. Even our metrics are more efficient than most previous metrics, they still face the problem of increasing annotated point number during the learning process. And the diversity metrics are typically more complicated than uncertainty metrics. Second, diversity metrics are not instructive in simple tasks (i.e., simple decision boundaries). Consider a binary classification task where the true decision boundary is just a line. Some points on the two sides of the line are enough to confirm where the line is. The informative points should be points with high uncertainty, which can help refine the current boundary. But diversity-based query strategies will continuously choose valueless points far away from the current boundary. And even in difficult tasks, we still expect uncertainty-based query strategies will perform better with enough queried points, since most points have been annotated[5]. To some extent, the tradeoff between uncertainty and diversity is the tradeoff between exploitation and exploration. Since the disadvantages of uncertainty and diversity metrics can be overcame by each other, we hypothesize combing the two metrics together can further accelerate learning.

3.3. Combination methods

Once we decide the uncertainty and diversity metrics, the next step is to combine them to generate a integrated metric. In this study, we consider three combination methods: Weighted arithmetic mean, weighted geometric mean and two stage query.

Weighted arithmetic mean Since all metrics are scaled to [0, 1], a intuitive way is to calculate the weighted arithmetic mean:

$$H(x) = \alpha U(\boldsymbol{x}) + (1 - \alpha) D(\boldsymbol{x})^6$$

The weight parameter α controls the relative importance of the uncertainty metric. Its range is [0, 1]. The hybrid metric degenerates to the uncertainty metric when α is 1 and degenerates to the diversity metric when α is 0.

In the above version, the weight is consistent during the whole learning process. But as we discussed above, the importance of diversity should decrease with more points queried. Thus, we also test a time-decayed version:

$$H(\boldsymbol{x}) = (1 - exp(-\beta t)) U(\boldsymbol{x}) + exp(-\beta t) D(\boldsymbol{x})$$

Here t represents the times of query having been done (including the current one, so the minimum is 1). The decay parameter β indicates how fast the weight of diversity decreases with query times. We also set its range to [0, 1].

Despite of its simplicity, weighted arithmetic mean is not popular in deep AL (Yin et al., 2017; Shui et al., 2019). One reason might be this combination is physically implausible

[5] Admittedly, this is contradictory since AL aims to use as less annotated points as possible. However, we confirm this speculation in the CIFAR-10 experiment.

[6] D here represents diversity rather than the discriminator score.

since each metric has its own unit. Consider an example of adding time and speed. This is meaningless though may be useful in some situations.

Weighted geometric mean Weighted geometric mean can create a more physically plausible metric and has been attempted by Li & Guo (2013). This calculation is

$$H(\boldsymbol{x}) = U(\boldsymbol{x})^{\alpha} D(\boldsymbol{x})^{(1-\alpha)}$$

Here the role of the weight parameter α is the same as that in constant weighted arithmetic mean. It also has the same range.

Similarly, we also test the time-decayed version of this combination method:

$$H(\boldsymbol{x}) = U(\boldsymbol{x})^{(1-exp(-\beta t))} D(\boldsymbol{x})^{exp(-\beta t)}$$

The role and range of the decay parameter β are also the same.

Two stage query The two stage query methods do not combine uncertainty and diversity metrics explicitly. The core idea is to use them successively. Suppose we will select m points in one query. Two stage query should first rank the points based on one metric and choose the top γm points, then choose m points from them based on another metric (Ash et al., 2021; Zhdanov, 2019). Since we can calculate the uncertainty and diversity metrics for each unannotated point. The selection at the second stage can be again simply choosing the top m points.

The two stage query in our study also has two variants: uncertainty-first and diversity-first. As the names suggest, the order of metrics used is uncertainty-diversity in the former variant but diversity-uncertainty in the latter variant. It also introduces a hyperparameter γ. It is an integer ranging from 1 to the pool size divided by m. The method will purely use the first metric when γ is 1, and purely use the second metric when γ is the pool size divided by m.

4. Experiments

4.1. MNIST experiment

One motivation of this experiment is to reproduce the results of Gal et al. (2017), which find the performance order of uncertainty-based query strategies is: variation ratio > BALD ≈ entropy. Another motivation is to compare the performances of query strategies in this simple task. We do not use the CNN architecture in the study of Gal et al. (2017) since some helpful tricks that can accelerate training, like batch normalization (Ioffe & Szegedy, 2015), are not considered. We finally adapt a architecture called Network in Network (NiN) (Lin et al., 2014). It entirely discards the fully-connected layers and has been shown powerful in mitigating overfitting[7].

[7]We have tried some other classical networks like VGG and ResNet, but they quickly overfit the data since AL considers few data points.

4.1.1. Implementation details

The model architecture in this experiment is shown in Figure 2. Since MNIST dataset is very simple, we apply an NiN with 5 layers (excluding batch normalization and dropout). Each image will be finally projected to a 10*1*1 tensor. This tensor will be flattened to a 10-dimensional vector, then passed to the softmax function to generate the predicted probabilities of classes.

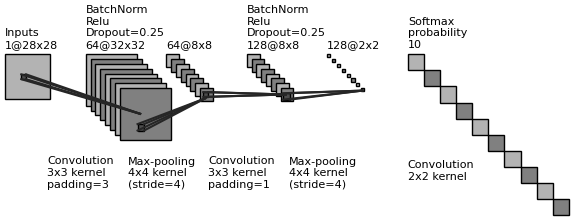

Figure 2. Model architecture for MNIST experiment

The output layer is also the embedding layer, i.e., the 10-dimensional vector is the low-dimensional representation of the corresponding image. It has two benefits. Firstly, each component of the vector directly shows the "support" of the image to the corresponding class. When the magnitude of each component is similar, diversity-based query strategies will tend to choose equal number of images from each class (based on current decision boundary). It can help get balanced samples, which is important in classification tasks. Secondly, if a constant is added to each component, the softmax outputs will not change. But diversity-based query strategies can capture this change and choose the more diverse point. The embeddings of points will be normalized before calculating the diversity.

At the training stage, 20 images are randomly selected from the training set as the initial training set. Each class contributes 2 of them. 1000 images are then randomly selected from the training set as the validation set. The rest images in the training set is the unannotated data pool. We set a batch size of 128 ("batch size" here indicates the number of points considered in one forward propagation), a epoch number of 50 and a learning rate of 0.001 using the Adam optimizer (Kingma & Ba, 2014).

After each training (i.e., 50 epochs), the model will query 10 points from the pool (to further reduce computational cost, 2000 points are randomly selected from the original pool as the actual pool) based on the corresponding query strategy (the forward propagation number at the prediction stage is 100 for BCNNs) and add them to the training set. Then it will be re-initialized and trained on the new training set. In each experiment, the model will conduct 20 times of query and test accuracy after each query will be recorded.

We run 5 experiments for each query strategy and average the results. Each experiment is controlled by a random seed (the random seeds are generated by a higher-level random seed specified before all experiments). It ensures the results

Query strategy	MNIST Acc.
Entropy	95.75
BALD	95.93
Variation ratio	97.01
Discriminator	93.52
Posterior Var	93.93
Min distance	95.70
Uniform	94.01
Time-decayed geometric mean of var ratio and posterior var	96.75

Table 1. Final test accuracy(%) on the MNIST experiment

are reproducible and results of different query strategies are comparable.

For the discriminator score metric, we built a small feed-forward neural network as the discriminator, whose architecture is shown in Figure 3. Its parameter settings are the same as those of the main BCNN.

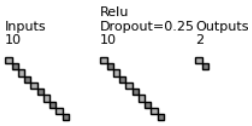

Figure 3. Model architecture for the discriminator.

For the first four hybrid query strategies, the hyperparameter ranges from 0 to 1. So we conduct a grid search from 0.1 to 0.9, with a step size of 0.1[8], and choose the value which produces the highest final mean validation accuracy (all details are the same as above, except replacing the test set with the validation set). For the latter two hybrid query strategies, we simply try 5 values of γ: 2, 3, 4, 5, 6, and choose the value using the same standard.

The pure query strategies together with a uniform selection are treated as baselines. So there are 7 baselines and 3 (uncertainty metrics) * 3 (diversity metrics) * 6 (combination methods) = 54 hybrid query strategies to be compared.

4.1.2. Results and discussion

Table 1 shows the final test accuracy for BCNNs with different query strategies. And Figure 4 shows the test accuracy as function of query times. Due to page limit, we only show the results of baselines and best hybrid query strategy. Performance of all hybrid query strategies can be found in the supplementary materials.

Our experiment basically reproduces the results of Gal et al. (2017) on the uncertainty-based query strategies. The performance order is still: variation ratio > BALD ≈ entropy. Besides, our model is more efficient since the uniform baseline reaches the accuracy of 94% using only 220 images,

[8]We have also tried a hyperparameter tuning using GPs, but it overfits quickly.

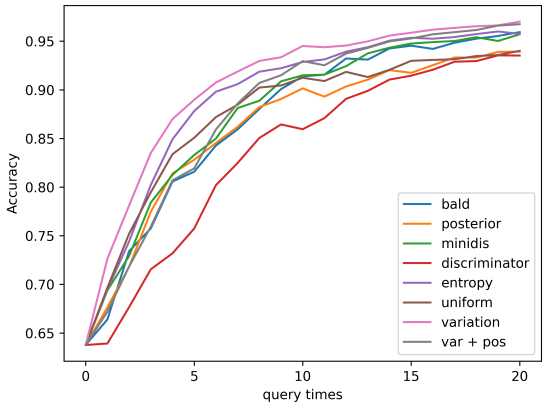

Figure 4. Test accuracy(%) as a function of query times on the MNIST experiment

while the accuracy is only about 88% with the same number of images in their study.

Surprisingly, all uncertainty-based query strategies outperform the uniform baseline, but only one diversity-based query strategy beats it. Considering uniform is also a diversity-based query strategy in general, diversity information may not be helpful in guiding query in this simple task. A further evidence comes from the best hybrid query strategy. It is the time-decayed weighted geometric mean of variation ratio and posterior variance. Its hyperparameter β is 0.6, indicating the weight of diversity metric will quickly decrease with query times. In the meantime, its performance is only comparable to the best baseline, and this pattern is pretty stable during the learning process (it is even beaten by the entropy query strategy at the early stage). All of those support uncertainty-based query strategies are enough to accelerate the learning in simple tasks.

However, MNIST is notorious for its simplicity and low-noise. Diversity-based query strategies are theoretically not suitable for such datasets. So the advantage of hybrid query strategies in combining different information may not be obvious in such situations. Thus, It is worth comparing the query strategies on a more complicated task.

4.2. CIFAR-10 experiment

The motivation of this experiment is to test whether task difficulty will adjust the performances of different query strategies, and whether hybrid query strategies can perform better in a difficult task.

4.2.1. Implementational details

The model architecture in this experiment is shown in the Figure 5. Considering the CIFAR-10 dataset is more challenging, we increase layer number from 5 to 10 and correspondingly decrease the dropout probability to accelerate training. The final output is still a 10*1*1 tensor, which will be flattened and passed to the softmax function. Also due to task difficulty, we increase the initial training set from 20 to 100 images. Each class contributes 10 of them. All other

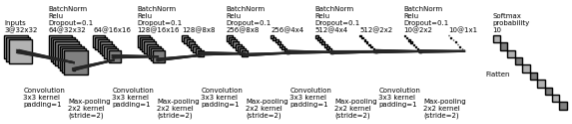

Figure 5. Model architecture for the CIFAR-10 experiment

Query strategy	Acc.	ExtendedExp Acc.
Entropy	41.04	67.91
BALD	40.80	66.47
Variation ratio	41.30	67.67
Discriminator	41.62	67.43
Posterior var	41.28	65.43
Min distance	42.48	65.15
Uniform	42.36	66.67
Time-decayed geometric mean of var ratio and min distance	43.60	67.62
Time-decayed arithmetic mean of var Ratio and posterior Var	43.52	68.18
Constant geometric mean of BALD and Min distance	43.53	66.64
Two stage diversity first of BALD and Discriminator	43.98	66.63

Table 2. Final test accuracy(%) on the CIFAR-10 experiment

settings are the same as those in the MNIST experiment.

After observing the results, we decide to run an "extended experiment" to see whether the advantages of hybrid query strategies will continue when more points are queried. This is because the final accuracy is relatively low and the points queried are still few compared to the dataset size. In real situations, the model should continuously query points until a threshold in accuracy is reached.

Specially, we choose the top 4 hybrid query strategies whose performances all surpass the best baseline. Along with 7 baselines, we compare 11 query strategies in this extended experiment. We increase the query times to 100 and number of points per query to 50. All other settings are the same as those in the main experiment, except we directly use the hyperparameter values tuned from the main experiment to decrease computational cost.

4.2.2. Results and discussion

Table 2 shows the final test accuracy for BCNNs with different query strategies on the main and extended experiments. And Figure 6 and 7 show the test accuracy as function of query times on the main and extended experiments. We show the results of baselines and top 4 hybrid query strategies since the performance differences among these hybrid query strategies are small.

The accuracy in the main experiment directly shows the larger difficulty of CIFAR-10. The models are more complex and use more images, but only reach accuracy lower than the half of the MNIST experiment. Meanwhile, the advantage of diversity-based query strategies in difficult tasks with limited queried points is supported. The best

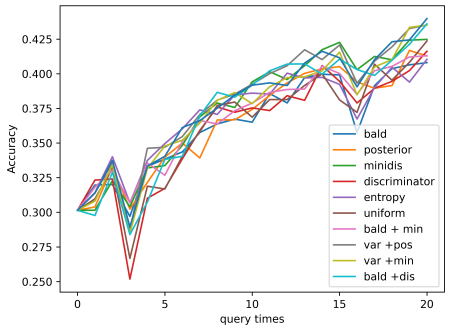

Figure 6. Test accuracy (%) as a function of query times on the CIFAR-10 experiment

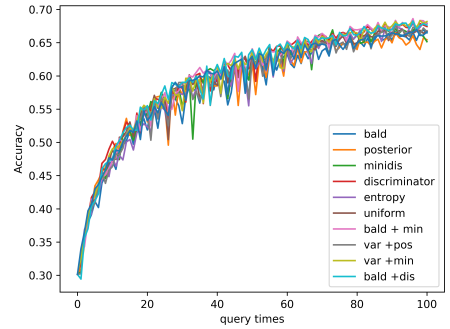

Figure 7. Test accuracy (%) as a function of query times on the CIFAR-10 extended experiment

uncertainty-based query strategy is only comparable to the worst diversity-based query strategy. The best diversity-based query strategy achieves a similar final accuracy with the uniform baseline, indicating diversity calculated between annotated and unannotated points can be as effective as that calculated within the unannotated points. Another finding is the hybrid query strategies shine in this situation. Several hybrid query strategies (see supplementary material for more instances) outperform the best baseline. This pattern is relatively stale during the learning process.

For the extended experiment, the position of uncertainty-based and diversity-based query strategies reverse again. However, the advantage of uncertainty-based query strategies is not obvious now, since its best member is only comparable to that of the diversity-based query strategies. The best uncertainty-based and diversity-based query strategies both outperform the uniform baseline, indicating the two information can both accelerate learning. The best query strategy is still hybrid, though its advantage compared to the best baseline decreases. From the accuracy curve, we can observe the performances of hybrid query strategies fluctuate less and are usually better than those of baselines.

Overall, the results support diversity-based query strategies are more efficient among pure query strategies in the difficult task with limited points queried, though this pattern is reversed with more queried points. Hybrid query strategies outperform pure query strategies in the difficult task. Their advantages decrease but still exist when more points are

queried.

5. Related work

Studies in applying AL to deep learning had not been extensive until recently. However, deep AL using hybrid query strategies is still an under-explored area. The main difference between our study and previous relevant studies is they typically do not calculate diversity metric for each point. We review some of them in this section.

To balance uncertainty and diversity, Ash et al. (2019) apply the gradient-based hybrid sampling technique using gradient embeddings to create diverse mini-batches of points. In this approach, the embedding of an unannotated point is the gradient of the last layer when adding it to the inputs (its label is assumed to be the current predicted label). Then a fixed-size determinantal point process (Kulesza & Taskar, 2011) will be used to select a mini-batch of points from the pool[9]. It will prefer points with high uncertainty (i.e., large gradient norm) when the batch size is small, but prefer points which are different from each other when the batch size is large. They compare it with other pure query strategies and show it consistently performs well or better regardless to batch sizes or architectures. The main contribution of this study is to introduce gradients as embeddings, which integrates uncertainty and diversity into a unified framework, and avoid hyperparameter tuning for query strategies. But conducting a determinantal point process over the pool is computationally costly.

Another example comes from Zhdanov (2019), who use a uncertainty-first two stage query to select points. This study tries different methods like K-means and nearest neighbours to construct a diverse sample of points at the second stage. It also compares the hybrid query strategies with a uncertainty-based query strategy in MNIST and CIFAR-10 datasets, and find K-means based hybrid query strategies always slightly outperform the baselines. It also finds the margin-based uncertainty metric is always better than the uniform baseline even in CIFAR-10 dataset, which may because it uses a much larger batch size than ours.

Yin et al. (2017) explicitly treat the tradeoff between uncertainty and diversity as the tradeoff between exploitation and exploration. They use entropy as the uncertainty metric. The diversity metric is the minimum minus dot product between a given point and points having been selected (see the second definition of diversity). The two metrics are combined using constant weighted arithmetic mean. To ensure the exploration will decrease when more points are queried, they first choose m points (m is smaller than the number of points per query) based on the uncertainty metric, then choose the rest points based on the hybrid metric. m will increase with query times until a threshold is reached. They compare it with some baselines on the MNIST dataset, and find it is always better than the pure query strategies. Interestingly, they also observe the uncertainty-based query strategies underperform the diversity-based query strategies at the early stage of learning, but surpass them when more points are queried. This is similar to what we find in the CIFAR-10 dataset. There are other studies (Hsu & Lin, 2015; Liu et al., 2018) directly modeling the switch between uncertainty and diversity in a reinforcement learning framework, but they are far away from our study.

[9]The fixed-size determinantal point process is mathematically equivalent to our posterior variance metric, though the diversity is calculated within the pool.

6. Conclusions

In this study, we propose two computationally efficient ways to calculate the diversity of unannotated points. Combining them with existing uncertainty and diversity metrics, we develop 54 hybrid query strategies for AL in a BCNN framework. The experiments comparing those query strategies show task difficulty and number of queried points can adjust the performances of query strategies. Uncertainty-based query strategies are enough to accelerate learning in simple tasks. When the task become difficult, diversity-based query strategies outperform uncertainty-based query strategies when limited points are queried, but the pattern is reversed when more points are queried. Hybrid query strategies perform better than pure query strategies in difficult tasks. Their advantages decrease but still exist with more queried points.

There are two promising directions to explore. The first is to decrease the training time of deep AL. We re-initialize the models after each query to isolate the effects of query strategies, but this is time-consuming and will be unaffordable with deeper model architectures and larger datasets. We have tried to only train the models on newly queried points without re-initialization, but the performances sharply decrease. We speculate this is because the points are not independent in this situation which breaks the assumption of stochastic gradient descent. One possible solution may be simply increasing batch size of query or decrease epochs to train the model. But a more reasonable solution may be using AL to finetune models (i.e., only few parameters are allowed to change) rather than train models from scratch. The second is to deign more ingenious hybrid query strategies. The best hybrid query strategies in experiments are usually time-decayed. And we empirically show the importance of diversity will decrease when more points are queried. Switching from diversity to uncertainty after the number of queried points reaches a threshold may be a more flexible hybrid query strategy. And as mentioned above, modeling the switch between uncertainty and diversity in a reinforcement learning framework may also help create a powerful hybrid query strategy.

References

Ash, Jordan T., Zhang, Chicheng, Krishnamurthy, Akshay, Langford, John, and Agarwal, Alekh. Deep batch active learning by diverse, uncertain gradient lower bounds. *CoRR*, abs/1906.03671, 2019. URL http://arxiv.org/abs/1906.03671.

Ash, Jordan T., Goel, Surbhi, Krishnamurthy, Akshay, and Kakade, Sham M. Gone fishing: Neural active learning with fisher embeddings. *CoRR*, abs/2106.09675, 2021. URL https://arxiv.org/abs/2106.09675.

Bontempelli, Andrea, Teso, Stefano, Giunchiglia, Fausto, and Passerini, Andrea. Learning in the wild with incremental skeptical gaussian processes. In *IJCAI International Joint Conference on Artificial Intelligence*. IJCAI, 2020. doi: 10.24963/ijcai.2020/399.

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

Freeman, Linton G. *Elementary applied statistics*. 1965.

Gal, Yarin and Ghahramani, Zoubin. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015a.

Gal, Yarin and Ghahramani, Zoubin. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2015b.

Gal, Yarin, Islam, Riashat, and Ghahramani, Zoubin. Deep bayesian active learning with image data. *CoRR*, abs/1703.02910, 2017. URL http://arxiv.org/abs/1703.02910.

Geifman, Yonatan and El-Yaniv, Ran. Deep active learning over the long tail. *arXiv preprint arXiv:1711.00941*, 2017.

Gissin, Daniel and Shalev-Shwartz, Shai. Discriminative active learning. *arXiv preprint arXiv:1907.06347*, 2019.

Hinton, Geoffrey E and Van Camp, Drew. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13, 1993.

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

Hsu, Wei-Ning and Lin, Hsuan-Tien. Active learning by learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Joshi, Ajay J, Porikli, Fatih, and Papanikolopoulos, Nikolaos. Multi-class active learning for image classification. In *2009 ieee conference on computer vision and pattern recognition*, pp. 2372–2379. IEEE, 2009.

Kingma, Diederik P and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, Alex, Hinton, Geoffrey, et al. Learning multiple layers of features from tiny images. 2009.

Kulesza, Alex and Taskar, Ben. K-dpps: Fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 1193–1200, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

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

Lin, Min, Chen, Qiang, and Yan, Shuicheng. Network in network. 2014.

Liu, Ming, Buntine, Wray, and Haffari, Gholamreza. Learning how to actively learn: A deep imitation learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1874–1883, 2018.

MacKay, David JC. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3): 448–472, 1992.

Nguyen, Vu-Linh, Destercke, Sébastien, and Hüllermeier, Eyke. Epistemic uncertainty sampling. In *Discovery Science: 22nd International Conference, DS 2019, Split, Croatia, October 28–30, 2019, Proceedings 22*, pp. 72–86. Springer, 2019.

Rasmussen, Carl Edward, Williams, Christopher KI, et al. *Gaussian processes for machine learning*, volume 1. Springer, 2006.

Ren, Pengzhen, Xiao, Yun, Chang, Xiaojun, Huang, Po-Yao, Li, Zhihui, Gupta, Brij B, Chen, Xiaojiang, and Wang, Xin. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

Sener, Ozan and Savarese, Silvio. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

Seung, H Sebastian, Opper, Manfred, and Sompolinsky, Haim. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294, 1992.

Shannon, Claude E. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Shui, Changjian, Zhou, Fan, Gagné, Christian, and Wang, Boyu. Deep active learning: Unified and principled method for query and training. *CoRR*, abs/1911.09162, 2019. URL http://arxiv.org/abs/1911.09162.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

Teso, Stefano and Vergari, Antonio. Efficient and reliable probabilistic interactive learning with structured outputs, 2022. URL https://arxiv.org/abs/2202.08566.

Tong, Simon. *Active learning: theory and applications*. Stanford University, 2001.

Yann, Lecun, Corinna, Cortes, and Christopher, Burges. The mnist database of handwritten digits. *The Courant Institute of Mathematical Sciences*, pp. 1–10, 1998. URL http://yann.lecun.com/exdb/mnist/.

Yin, Changchang, Qian, Buyue, Cao, Shilei, Li, Xiaoyu, Wei, Jishang, Zheng, Qinghua, and Davidson, Ian. Deep similarity-based batch mode active learning with exploration-exploitation. In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 575–584. IEEE, 2017.

Zeni, Mattia, Zhang, Wanyi, Bignotti, Enrico, Passerini, Andrea, and Giunchiglia, Fausto. Fixing mislabeling by human annotators leveraging conflict resolution and prior knowledge. 2019. doi: 10.1145/3314419. URL https://doi.org/10.1145/3314419.

Zhdanov, Fedor. Diverse mini-batch active learning. *arXiv preprint arXiv:1901.05954*, 2019.

Supplementary materials

Uncertainty	Diversity	MNIST Acc.	CIFAR Acc.
Entropy	Discriminator	96.36	41.73
	Posterior Var	96.55	40.34
	Min distance	96.26	42.63
BALD	Discriminator	96.06	39.14
	Posterior Var	96.40	38.38
	Min distance	95.71	43.53
Var ratio	Discriminator	96.59	42.43
	Posterior Var	94.62	42.66
	Min distance	96.19	42.21

Table 3. Constant weighted geometric mean

Uncertainty	Diversity	MNIST Acc.	CIFAR Acc.
Entropy	Discriminator	95.64	40.54
	Posterior Var	96.09	40.34
	Min distance	96.26	43.14
BALD	Discriminator	96.15	40.27
	Posterior Var	96.05	40.98
	Min distance	95.65	43.27
Var ratio	Discriminator	96.73	41.66
	Posterior Var	96.75	41.45
	Min distance	96.59	43.60

Table 4. Time-decayed weighted geometric mean

Uncertainty	Diversity	MNIST Acc.	CIFAR Acc.
Entropy	Discriminator	96.28	41.22
	Posterior Var	96.35	40.70
	Min distance	96.20	41.80
BALD	Discriminator	95.83	41.68
	Posterior Var	95.93	41.13
	Min distance	96.21	43.05
Var ratio	Discriminator	96.38	42.00
	Posterior Var	96.50	42.39
	Min distance	96.35	42.96

Table 5. Constant weighted arithmetic mean

Uncertainty	Diversity	MNIST Acc.	CIFAR Acc.
Entropy	Discriminator	96.30	42.68
	Posterior Var	96.51	41.99
	Min distance	96.55	41.42
BALD	Discriminator	95.84	40.16
	Posterior Var	95.87	42.45
	Min distance	96.02	42.87
Var ratio	Discriminator	96.62	40.14
	Posterior Var	96.58	43.52
	Min distance	96.60	42.21

Table 6. Time-decayed weighted arithmetic mean

Uncertainty	Diversity	MNIST Acc.	CIFAR Acc.
Entropy	Discriminator	93.72	41.31
	Posterior Var	94.22	41.71
	Min distance	94.34	42.06
BALD	Discriminator	93.71	41.01
	Posterior Var	94.24	40.72
	Min distance	94.40	42.94
Var ratio	Discriminator	93.80	39.59
	Posterior Var	94.26	42.01
	Min distance	94.78	42.57

Table 7. Two stage query-uncertainty first

Uncertainty	Diversity	MNIST Acc.	CIFAR Acc.
Entropy	Discriminator	93.52	41.39
	Posterior Var	94.23	41.34
	Min distance	94.27	41.88
BALD	Discriminator	94.25	43.98
	Posterior Var	94.23	40.28
	Min distance	94.25	43.37
Var ratio	Discriminator	94.39	41.01
	Posterior Var	94.41	42.91
	Min distance	94.28	42.09

Table 8. Two stage query- diversity first