Deep Active Learning Through Cognitive Information Parcels

Wencang Zhao
1. College of Automation and Electronics Engineering, Qingdao University of Science and Technology, Qingdao, Shandong, China 266061
2. Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA 02115
wencangzhao@gmail.com

Zhengming Ding
Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA 02115
dingz312@ece.neu.edu

Yu Kong
Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA 02115
yu Kong@ece.neu.edu

Yun Fu
1. Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA 02115
2. College of Computer and Information Science, Northeastern University, Boston, MA, USA 02115
yunfu@ece.neu.edu

ABSTRACT
In deep learning scenarios, a lot of labeled samples are needed to train the models. However, in practical application fields, since the objects to be recognized are complex and non-uniformly distributed, it is difficult to get enough labeled samples at one time. Active learning can actively improve the accuracy with fewer training labels, which is one of the promising solutions to tackle this problem. Inspired by human beings’ cognition process to acquire additional knowledge gradually, we propose a novel deep active learning method through Cognitive Information Parcels (CIPs) based on the analysis of model’s cognitive errors and expert’s instruction. The transformation of the cognitive parcels is defined, and the corresponding representation feature of the objects is obtained to identify the model’s cognitive error information. Experiments prove that the samples, selected based on the CIPs, can benefit the target recognition and boost the deep model’s performance efficiently. The characterization of cognitive knowledge can avoid the other samples’ disturbance to the cognitive property of the model effectively. We believe that our work could provide a trial of thought about the cognitive knowledge used in deep learning field.

CCS CONCEPTS
• Computing methodologies → Neural networks;

KEYWORDS
deep learning, active learning, cognitive information

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 ACM. ISBN 978-1-4503-4906-2/17/10... $15.00
DOI https://doi.org/10.1145/3123266.3123337

1 INTRODUCTION
At present, deep learning remains a fast developing field with many research areas under explored [14, 31, 36]. Since the deep models have more layers, they need a great many samples to be trained to get the ideal performance [30, 44]. However, in real-world applications, especially the industrial detection or medical areas [18], the recognized objects are complex and non-uniformly distributed [4, 33, 40]. It is difficult or expensive to obtain a large amount of labeled samples initially [12, 35, 55].

Some of the solutions can be found by contrasting current deep learning methods to other kinds of learnings we observe in natural systems such as humans and other animals [1, 18, 33, 41]. But human-beings’ learning procedures have rarely been seen in machine learning mechanism [3, 37, 50]. When human-beings learn new knowledge, they often select a few examples under the instruction from a teacher gradually [37]. The illustration of the cognitive process imitating human-beings is shown as Figure 1. In addition, if deep models are trained with more labeled data, they could exhibit human-like learning properties that mirror those of humans [9, 30]. So, a practical solution is to train a deep model using a very small number of samples initially to improve the model’s performance through adding more effective samples to the training set under the expert’s instruction actively [7, 15, 52]. That is to say, when new objects or patterns are to be learned, the samples which will be mostly misclassified are selected as the training data according to the cognition of the model [39].

For the sake of imitating the implementation as Figure 1 shows based on deep learning method, the deep models can be trained with the dropout method [7, 47] using a small number of labeled samples at the beginning. Moreover, deep models should be able to distinguish the confident and uncertain objects, and the uncertainty value can be calculated by averaging the stochastic forward passes through the models [10, 11, 46]. Furthermore, there are still two other problems that should be solved [15, 49]. First, the models can select the most effective samples to add to the training set under the experts’ help. Second, the deep models can enhance the recognition
A new active learning framework is proposed, which can be used in practical application /field with deep models. 3) The cognitive errors are used as the learning materials.

In order to select the most effective samples, we propose the Cognitive Information Parcels (CIPs), combining the deep models’ cognitive errors and the instruction from the expert. Here, the CIPs are a kind of feature, which synthetically merge the three cognitive errors and the instruction from the expert. Since the errors are generated by the models, the biggest CIPs value reflects the maximum confusion from the models on the data. The most confusing examples are selected as the Target-Sensitive Samples (TSSs) (i.e. effective samples) and added into the training dataset to improve the models’ performance. The feature characterizes the cognitive errors comprehensively between the models’ prediction and the ultimate target. It has the better results than the variation ratios, predictive entropy, and Bayesian Active Learning by Disagreement (BALD) methods [10, 12, 19], which select samples based on uncertainty information. For the sake of improving the performance of the deep models gradually, we separate the whole procedure into two stages, the working stage and the updating stage, according to the need of practical application conditions. During the working stage, the model can give the recognition results on the confident input, and put the uncertainty examples into uncertain pool to seek instruction from the expert. In the updating stage, the most puzzling examples are used as the training data to boost the model.

In summary, the innovations of this paper are: 1) The cognitive information parcels are designed, which are novel features used to select effective samples to boost the deep models’ performance. 2) A new active learning framework is proposed, which can be used in practical application field with deep models. 3) The cognitive knowledge is used in deep learning, which can decrease the deep models’ cognitive errors effectively.

2 RELATED WORK

Aiming at solving problems in the practical applications and inspired by human-beings’ learning process [5], we propose a method to select effective samples to finely tune the deep models actively. Since this work is so challenging that there are very limited literatures which focus on active image recognition based on deep models [12, 21, 28, 29, 34].

2.1 Active learning

The primary idea in active learning is that a model actively acquires higher accuracy by fewer examples if it is permitted to use the samples from which it is learning [6, 51]. An active model might apply for queries, and usually an oracle labels the data. Active method is well-motivated in numerous learning areas, where unlabeled examples might be abundant or acquired easily, but labeling job is difficult, time-consuming or expensive [8, 16].

There have been several methods to formulate query. The most commonly used query method is uncertainty sampling [10]. In this way, an active model queries samples on which it is the least certain. However, the idea for the least confident method only uses the information on the most probable sample, and it casts the information on the distribution of the other labels. To correct the deficiency, the multi-class uncertainty sampling variant [42], marginal sampling was developed. Nevertheless, for problems with large labeled sample sets, the marginal approach still neglects much of the information of remaining classes. Afterwards, bayesian active learning by disagreement method [19] was proposed based on the theory of mutual information. Apart from these, Burr and Mark [43] proposed the information density, sequence vote entropy and Fisher information query by committee. Feras et al. [7] proposed an active learning method based on episode with Bayesian neural networks. But these query methods do not consider the sample selection method like human-beings’ cognitive way.

2.2 Deep learning

Deep learning achieves the state-of-the-art performance in many machine learning tasks, but we do not have a strong understanding of what or how deep models learn [9, 25, 48]. Lalor et al [30] compared deep models’ performance with human directly, they found deep models exhibit human-like learning properties, and different models exhibit different strengths in learning. However, these query methods do not consider the sample selection method like human-beings’ cognitive way.
Figure 2: This framework shows the implementation of the deep active learning. Firstly, train an initial model using fewer labeled examples. During the working stage (blue frame), the model could generate the confident recognition on the familiar objects; otherwise, the model would seek help from the expert to analyze the uncertain samples, and the expert gives the instruction based on analyzing the model’s errors. Then, during the updating stage (green frame), the Cognitive Information Parcels (CIPs) are calculated by evaluating the cognitive errors and the Target Sensitive Samples (TSSs) are selected. Finally, the TSSs are added into the training set to finely tune the deep model and such procedure continues.

method, which does not accord with the human-beings’ learning procedure [46]. Moreover, humans recognize a new object from coarse to fine, from complex to simple [15]. It is significant to train the deep models referencing human-beings’ related properties to improve the deep models’ performance. The long-term goals for machine learning are to make the machine learn and think like human-beings’ doing [53, 54], and the human cognitive capabilities or human-like cognitive models should be introduced into the intelligent systems to develop a new form of learning mechanism [18], such as hybrid-augmented intelligence [56].

As far as we know, the existing literatures seldom discuss the state-of-the-art about the field in detail [26]. We attempt to design a novel deep active learning by choosing the effective samples to finely tune a deep model actively. But how to design a mechanism like human-beings’ doing by mathematics puzzles the researchers recently [17, 20], especially in deep learning areas.

3 DEEP ACTIVE LEARNING BASED ON COGNITIVE INFORMATION PARCELS

In the section, we formulate a novel framework in the light of real-world applications through deep active learning method, and develop our approach systematically. Then the target-sensitive sample selection algorithm is proposed. The framework is shown in Figure 2 and the notations used throughout the paper are summarized in Table 1.

3.1 Motivation

Our goal is to improve the deep models’ performance through a few effective training samples like human-beings’ learning gradually.

To achieve the goal, we develop this deep active learning framework shown in Figure 2. A deep model is trained by a small number of labeled examples initially. Since the objects are complex or the working environment changes frequently, the samples are non-uniformly distributed. In the working stage, the model can give the ultimate recognition results on the confident inputs. Otherwise, for the uncertain ones, the model seeks help from the expert who gives the related instruction and helps the model to produce the ultimate results. During the model updating period (i.e. working clearance or appointed time), the model transforms the cognitive errors on the uncertain samples to cognitive parcels under the expert’s instruction and selects the most effective samples as the training data to enhance the model’s performance.

3.2 Confidence of deep models

In our deep active learning scenarios, the primary job is to achieve the confidence informativeness of the deep model on input samples. There are several ways to formulate such kind of strategy in confidence calculating literatures. If a network is trained by placing a probability distribution over each weight $\omega$, it obtains a predictive mean $E[y]$, and a predictive variance $Var[y]$ on the input $x$, which means how much the model is confident on its prediction [10]. Here, we adopt three frequently-used methods which could be used to evaluate the confidence property for deep active learning framework.

Deep variation ratio property. To calculate the variation ratio information of a deep recognition model’s output [12], a label should be sampled from the soft-max probabilities at the end of each stochastic forward pass from an input sample $x$. A set of $T$ labels $\hat{y}_t(x) \in \{\hat{y}_1(x), \cdots, \hat{y}_T(x)\}$ are collected from multiple stochastic forward passes on the same input $x$. Then the variation ratio can be calculated as:

$$U(x) = 1 - \frac{\ell_m(x)}{f}.$$  

Table 1: Notations and descriptions

<table>
<thead>
<tr>
<th>Variable Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{y}(x)$</td>
<td>Prediction of the model on input $x$</td>
</tr>
<tr>
<td>$X_{PE}$</td>
<td>Prediction error of the model</td>
</tr>
<tr>
<td>$H_{VP}$</td>
<td>Change information of prediction on class</td>
</tr>
<tr>
<td>$H_{VC}$</td>
<td>Prediction variation between different classes</td>
</tr>
<tr>
<td>$\bar{y}(x)$</td>
<td>Class prediction of input $x$</td>
</tr>
<tr>
<td>$S_{TS}$</td>
<td>Target-sensitive sample</td>
</tr>
<tr>
<td>$U(x)$</td>
<td>Uncertainty value</td>
</tr>
<tr>
<td>$\beta_m$</td>
<td>Cognitive information parcels</td>
</tr>
<tr>
<td>$S_{U}$</td>
<td>Uncertainty sample</td>
</tr>
<tr>
<td>$U_p$</td>
<td>Uncertainty sample pool</td>
</tr>
<tr>
<td>$T_S$</td>
<td>Training dataset</td>
</tr>
<tr>
<td>$C_m$</td>
<td>Cognitive feature domain</td>
</tr>
<tr>
<td>$T$</td>
<td>Number of stochastic forward passes</td>
</tr>
<tr>
<td>$C$</td>
<td>Class domain</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Model weight</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Feature restriction parameter</td>
</tr>
<tr>
<td>$\omega_t$</td>
<td>Model parameter at the time $t$</td>
</tr>
<tr>
<td>$c$</td>
<td>Class that the sample belongs to</td>
</tr>
</tbody>
</table>
Deep entropy property. Unlike variation ratio property, the predictive entropy of the deep models has its foundations in information theory [12, 45], which is shown as follows:

$$H[y|x] = -\sum_c p(y = c|x) \log p(y = c|x), \quad (2)$$

summing over all possible classes $c$ that $y$ can take. Given a test point $x$, the predictive entropy attains its maximum value when all classes are predicted to have equal uniform probability, and its minimum value when one class has probability 1 and with all other probability 0 (i.e. the prediction is certain).

Deep BALD property. As a improvement to the predictive entropy property, based on the mutual information between the prediction $y$ and the posterior over the model weight $\omega$, the Bayesian Active Learning by Disagreement (BALD) [10, 19] can be constructed as follows,

$$U(x) = H[\frac{1}{T} \sum_{t=1}^{T} p(y_t|x_t)] - \frac{1}{T} \sum_{t=1}^{T} H[p(y_t|x_t)] \quad (3)$$

where $H[\frac{1}{T} \sum_{t=1}^{T} p(y_t|x_t)]$ is the entropy of the average predicted probability, i.e., the model seeks the samples about which the model is most uncertain about the average output. The second term in $U(x)$ given by $\frac{1}{T} \sum_{t=1}^{T} H[p(y_t|x_t)]$ seeks the samples for which the average uncertainty is low.

The variation ratio provides a measure of how “spread” the output distribution is around the class and the entropy could capture the average amount of information contained in the predictive distribution. As comparison, the deep BALD method offers the sharing information between object $x$ and its output $\hat{y}(x)$, which contains more comprehensive information about the model on the input samples than the former two. In the paper, we adopt the deep BALD to calculate the model’s confidence on the input objects.

3.3 Cognitive information parcels

While the deep models are trained with more data, they exhibit human-like learning properties that reflect those of humans [30], which means that the deep models’ learning is similar to human’s learning to some extent [9]. In human-beings’ learning procedure, how to select effective learning materials to boost the learning efficiency is essential [27]. However, in deep learning field, tons of labeled data are needed to train the deep models, which is incongruous with human’s learning [1]. In order to explore the sample’s influence to deep models’ training, we design a novel active learning framework, where one of the important jobs is to evaluate the information of the uncertain samples to select more effective training data based on the instruction from expert.

As Figure 1 shows, when a learner encounters a new object, he would be confused and give an uncertain prediction about it. The uncertainty contains three factors: the error of the object’s prediction, the error variation of the object’s prediction, and the prediction change between different classes. Now the teacher as an expert could give help, provide some instructions based on the analysis of the learner’s uncertainty, and give some effective examples to learn. The prediction-instruction-selection (effective samples)-learning procedure is a kind of human-beings’ cognitive process, which is a closed loop and mutual interaction process between the learner and teacher [15]. This is a recurrent comparing and weighing procedure to understand the new object or knowledge from coarse to fine, from complex to simple [17].

Inspired by the learning process of human-beings, we develop the Cognitive Information Parcels (CIPs) combining the models’ prediction and the experts’ instruction. Then, considering the real-world applications of the deep models in working fields, the sample selection method is proposed to improve the models’ recognition accuracy actively. Since Bayesian inference is a perfect tool to analyze deep learning [13], we give the following several descriptions along with the CIPs using the statistical method. As Figure 1 shows, the human-beings’ uncertainty is reflected in three aspects. To train the model like human-beings do, all the above uncertainty factors should be considered in a kind of hybrid way.

For cases with more than two class labels, an uncertain variable related to the first factor, the prediction error $(PE)$, can be described as, $X_{PE} = 1 - p_{\omega}(\hat{y}(x))$, here, $\hat{y}$ is the class label with the instruction from an expert under the model’s current weight $\omega$. Another way to compute this kind of error is using the expected 0/1 loss, such as the belief of the model, which will mislabel the input $x$. The second uncertainty factor is the variation of the prediction $(VP)$ on the most probable class after different predicted times, $VP_T = X_{PE} - X_{PE_{t-1}}$. The difference between the $t$ time’s $X_{PE}$ and the $(t-1)$ time’s $X_{PE_{t-1}}$, which reflects the instability of the $X_{PE}$. We evaluate the model using the Stochastic Regularization Techniques (SRT) to obtain a random output through different stochastic forward passes, which is repeated several times (here $T$ repetitions), sampling the outputs $(\hat{y}_1(x), \cdots, \hat{y}_T(x))$. For obtaining the predicted classes’ probabilities $p(\hat{y}(x))$, we use the deep CNN model with dropout after every parameter layer. Then we average $T$ stochastic forward passes through the model [32]. The experiments demonstrate that this approach can outperform simply calculating the predictive probabilities of a single pass through the model.

The above criterion considers the uncertainty of the predicted error’s variation on the probable label. In order to obtain the informativeness, we use the entropy method to calculate the related information quantity. We evaluate $T$ times $VP$S, and average these entropy quantities to describe the $VP$’s change information as $H_{VP} = -\frac{1}{T} \sum_{t=1}^{T} VP_T \log VP_T$. Concerning the third uncertain change factor, i.e. the prediction variation on different classes $(VC)$, we also use the above method to average $T$ time’s entropy quantities between $C$ classes as follows:

$$H_{VC} = -\frac{1}{T} \sum_{j=1}^{T} \frac{1}{C} \sum_{i=1}^{C} p_{ij}(y_{ij}|x) \log p_{ij}(y_{ij}|x). \quad (4)$$

here, $T$ is the repetition times of stochastic forward passes, $C$ is the class label number and the $y_{ij}$ is the related output of $y_i$ in $T$ stochastic forward passes. When a sample $x$ is inputted into the model, the model will give the prediction values of all other classes beside the most probable one, and the $H_{VC}$ reflects the change information between the different classes.

Then, the above three uncertain factors are synthetically considered and the cognitive information parcel $\beta_x$ can be drawn forth as
Uncertain Images

\[ \beta_1 \newline \beta_2 \beta_3 \newline \beta_4 \beta_5 \beta_6 \newline \beta_7 \beta_8 \beta_9 \newline \beta_{10} \newline \beta_{11} \beta_{12} \newline \beta_{13} \beta_{15} \beta_{14} \newline \beta_4 \newline \beta_5 \newline \beta_8 \newline \beta_{11} \newline \beta_{12} \newline \]

Select Calculate HVP
TSSs
CIPs Values
\[ \beta_m \]

where uncertain factors of the current model property comprehensively, it learning procedure. Since this kind of feature considers all the and characterizes the cognitive knowledge occurring in the model abilities of the current models’

The feature could effectively represent the classi-

By computing the CIPs \( \beta_{m} \) that comprehensively merge three cognitive parcels \( \{X_{FE}, H_{VP}, H_{VC}\} \) as the target-sensitive samples and add them into the training dataset \( T_S \) to fine-tune the deep models during the model updating period. The detailed pseudo of the target-sensitive samples’ generation process is presented in Algorithm 1.

4 EXPERIMENTS AND RESULTS

4.1 Datasets and experimental settings

Since there is no proper industrial or medical labeled dataset to be used now, for the sake of evaluating the nature of our method in the paper, we choose handwriting digits’ dataset, i.e. MNIST, as the test database. Because different persons have various writing habits, the same character written by different persons has multifarious appearances. So the MNIST dataset is one of the best databases for researcher who wants to test learning techniques and pattern recognition ideas on the real application data, which is complex and non-uniformly distributed. Moreover, in order to prove the effects of background change, we choose CIFAR-10, CIFAR-100 databases to demonstrate the influence of the input images’ complexity on the model’s properties. By comparison, since the later two datasets have 10 and 100 classes’ color images with different shapes and backgrounds respectively, the complexity of these three datasets increases from MNIST to CIFAR-10 and CIFAR-100 gradually. The above three datasets are all non-uniformly distributed and usually employed in the computer vision fields.

In our paper, the MNIST, CIFAR-10 and CIFAR-100 training datasets are divided into initial training batched with 10,000, 5,000 and 5,000 images, validation batches with 5,000, 5,000 and 5,000 images and pool data batches with 40,000, 40,000 and 40,000 images respectively. The pool data images are used for creating the uncertainty pool to produce the target-sensitive samples to train the deep models actively.

Unless specified, we use the below optimal settings for all the experiments. The experiments are implemented with 9 layers dropout

Algorithm 1: Selecting the target-sensitive samples (TSSs)

Input: Samples \( S_U \) from uncertain sample pool \( U_p \)

for Samples \( S_U \) do

for \( T \) stochastic forward passes do

Calculate \( P(y|x) \) on \( S_U \)
Calculate \( X_{FE} \)
Calculate \( H_{VP} \)
Calculate \( H_{VC} \)

end for

Calculate \( \beta_m \)
Select \( S_{TS} \)
Add \( S_{TS} \) into training dataset \( T_S \)

end for

Output: Target-sensitive samples (TSSs)
CNN [12] on the above three datasets. To be sure, the other types of deep models can be employed and get the similar results. The 20 images, 200 images and 200 images are selected as the initial training data for MNIST, CIFAR-10, and CIFAR-100 to train the initial models respectively. Likewise, during the updating stage, 10 images, 200 images and 200 images are selected from the uncertainty pool through the effective samples choosing method and added into the training dataset to improve the models according to the corresponding database separately.

4.2 Results and analysis
To evaluate the nature of our method in the paper, denoted as CIPs, we compare the CIPs with the other three methods: the variation ratios [12], predictive entropy [12] and bayesian active learning by disagreement [10, 19] (abbreviated as VarRatio, Entropy and BALD respectively). The four methods select training data to finely tune the models in the same conditions. We run the experiments on the above three databases: MNIST, CIFAR-10 and CIFAR-100 to test the models when the complexity of the recognized objects is increasing. The results and the corresponding analysis of the selected training samples, model accuracy, validation loss and recognition results will be discussed below.

Selected training samples. The first four selected samples, which are used as the training data by the above four methods, are shown in Figure 4. The left block is from the MNIST database, the middle one is on the CIFAR-10, and the right one is from the CIFAR-100 database. In each block, the (a) to (d) columns denote the samples selected by the algorithms: VarRatio, Entropy, BALD and CIPs in turn. From the displayed images, we can see that different methods select the diverse training samples from the uncertainty pool to finely tune the models. The results disclose that the CIPs method prefers the more intangible images as the effective ones, obviously to the MNIST database, the CIPs-10 database takes second place and the CIFAR-100 is the most unapparent. The following experiment results show that different algorithms will get different recognition accuracy and the results from the CIPs method are the best ones on all the three databases, especially to the MNIST database. So, it is necessary to design the reasonable algorithm to finely tune the deep models to get better recognition results through selecting effective training samples.

Model test accuracy. Figure 5 shows the test accuracy of four methods based on the three databases. From the results, we can see that our method, CIPs, is better than the other three ones (i.e. BALD, Entropy and VarRatio) from the start to the last tuning iteration on all of the experimental databases. Moreover, we can find that the stability of the CIPs is better than the others’ also, even if the fluctuations are increasing when the complexity of the objects is raising from database MNIST to CIFAR-10, even to CIFAR-100. The experimental results discover that the fluctuation in recognition results raises when the complexity of the objects is increasing and it needs more tuning iterations to reach the stable outcomes. In addition, the MNIST results suggest that the CIPs method reaches the stable high recognition accuracy nearly after the 20th tuning iteration, but the other three methods need more than 40 tuning iterations to reach stability. Furthermore, from the results on three databases, we see that the Entropy and the BALD methods have the nearly similar properties because the latter uses mutual information to gain the uncertain informativeness which is homogeneous to the former. Nevertheless, the variation ratio method is well-behaved sometimes at the early learning stage.

Model validation loss. The model validation losses from four methods are shown as Figure 6. In order to gain the convergence on different complex data, here, we run 50 epoch iterations on MNIST database and 200 epoch iterations on the other two databases during the training time. Considering the results from MNIST, the bars show that the CIPs converges from the 10th epoch, the BALD and Entropy methods converge after the 40th epoch and the VarRatio needs more epochs to converge. The later two figures (b and c) demonstrate that all of the methods need more iterations of epoch on CIFAR-10 and CIFAR-100 databases to converge. However, all the results suggest that our method, CIPs, converges faster than the other three ones on three databases. When the objects are becoming complex as from MNIST to CIFAR-10 and then to CIFAR-100, the models would need more epoch iterations to converge during the fine-tuning time.

Model recognition results. Tables 2 to 4 show the model recognition results on database MNIST, CIFAR-10 and CIFAR-100 respectively. In order to compare conveniently, we select 13 tuning times from 4th to 100th every 8 iterations and list the results in the related tables. The results with bold font in every table indicate that our method’s recognition results are better than the other methods’ in all the tuning iterations. The CIPs’ results on the MNIST database show that the model gets outstanding recognition results after few tuning iterations, which reaches the recognition rate over 99% after the 68th tuning. The results in tables 3 and 4 demonstrate that the complex examples in CIFAR-10 and CIFAR-100 demand more learning iterations to reach the same higher recognition results, which exactly likes the human-beings’ learning also, i.e. difficult knowledge requires more times and endeavors to learn; and the recognition rates are not ideal in the less tuning iterations especially on CIFAR-100 database.

To summarize, from the above experiments we can see that our method is superior to other three ones on three databases. If the objects are more complex or pattern changing tremendously, the models need more learning times to be adapted to the intricate
Figure 5: The test accuracy comparison of four different methods on three databases. The results tell that the CIPs method is better than the others on the accuracy, the stability and the tuning iteration where the method reaches stable result. Along with the increasing of the complexity of the objects, i.e. from the database MNIST to CIFAR-10, and even to CIFAR-100, the test accuracy becomes worse. But the performance is improved with the raise of the learning times.

Figure 6: The validation loss comparison of different methods. The results suggest that the convergence nature of CIPs is better than the other methods on three databases. If the data are more complex, like the CIFAR-10 and CIFAR-100, more epochs are needed to converge and the loss values become bigger also.

Table 2: Recognition results(%) on MNIST database. The results show that the CIPs method gets higher recognition results in all the tuning iterations and reaches the rate 99% from the 68th tuning. The BALD and the Entropy methods have the nearly same results in the most of iterations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Train4</th>
<th>Train12</th>
<th>Train20</th>
<th>Train28</th>
<th>Train36</th>
<th>Train44</th>
<th>Train52</th>
<th>Train60</th>
<th>Train68</th>
<th>Train76</th>
<th>Train84</th>
<th>Train92</th>
<th>Train100</th>
</tr>
</thead>
<tbody>
<tr>
<td>VarRatio</td>
<td>59.76</td>
<td>79.29</td>
<td>86.61</td>
<td>88.29</td>
<td>92.06</td>
<td>92.59</td>
<td>93.04</td>
<td>94.08</td>
<td>94.29</td>
<td>95.02</td>
<td>95.11</td>
<td>95.22</td>
<td>95.59</td>
</tr>
<tr>
<td>Entropy</td>
<td>61.19</td>
<td>76.86</td>
<td>87.83</td>
<td>90.15</td>
<td>94.19</td>
<td>95.34</td>
<td>96.30</td>
<td>96.89</td>
<td>97.43</td>
<td>97.75</td>
<td>97.85</td>
<td>97.98</td>
<td>98.20</td>
</tr>
<tr>
<td>BALD</td>
<td>53.57</td>
<td>79.45</td>
<td>90.24</td>
<td>91.69</td>
<td>95.09</td>
<td>96.22</td>
<td>96.39</td>
<td>96.95</td>
<td>97.16</td>
<td>97.77</td>
<td>98.06</td>
<td>97.76</td>
<td>98.18</td>
</tr>
<tr>
<td>CIPs</td>
<td>68.72</td>
<td>95.41</td>
<td>97.56</td>
<td>98.23</td>
<td>98.61</td>
<td>98.68</td>
<td>98.89</td>
<td>98.98</td>
<td>99.00</td>
<td>99.05</td>
<td>99.04</td>
<td>99.13</td>
<td>99.12</td>
</tr>
</tbody>
</table>

Table 3: Recognition results(%) on CIFAR-10 database. The results from the CIPs method have higher recognition rates from the start to the last tuning iteration and the other three methods have nearly the same results in the most of iterations. In the earlier iterations the Entropy method has the better results than the BALD, but after the 28th tuning iteration the BALD has the higher ones than the Entropy.

<table>
<thead>
<tr>
<th>Method</th>
<th>Train4</th>
<th>Train12</th>
<th>Train20</th>
<th>Train28</th>
<th>Train36</th>
<th>Train44</th>
<th>Train52</th>
<th>Train60</th>
<th>Train68</th>
<th>Train76</th>
<th>Train84</th>
<th>Train92</th>
<th>Train100</th>
</tr>
</thead>
<tbody>
<tr>
<td>VarRatio</td>
<td>35.92</td>
<td>52.41</td>
<td>57.59</td>
<td>63.26</td>
<td>66.16</td>
<td>68.05</td>
<td>70.19</td>
<td>71.11</td>
<td>71.68</td>
<td>73.02</td>
<td>72.65</td>
<td>73.92</td>
<td>75.28</td>
</tr>
<tr>
<td>Entropy</td>
<td>36.83</td>
<td>53.26</td>
<td>53.78</td>
<td>61.72</td>
<td>64.47</td>
<td>67.22</td>
<td>68.46</td>
<td>69.90</td>
<td>71.37</td>
<td>73.19</td>
<td>72.66</td>
<td>74.12</td>
<td>75.67</td>
</tr>
<tr>
<td>BALD</td>
<td>36.77</td>
<td>52.70</td>
<td>55.72</td>
<td>63.38</td>
<td>66.08</td>
<td>68.44</td>
<td>69.71</td>
<td>70.95</td>
<td>72.83</td>
<td>73.24</td>
<td>73.19</td>
<td>74.93</td>
<td>76.00</td>
</tr>
<tr>
<td>CIPs</td>
<td>40.49</td>
<td>58.24</td>
<td>63.50</td>
<td>68.08</td>
<td>70.81</td>
<td>72.70</td>
<td>74.64</td>
<td>76.23</td>
<td>76.86</td>
<td>77.78</td>
<td>78.01</td>
<td>78.41</td>
<td>78.92</td>
</tr>
</tbody>
</table>

conditions. To demonstrate this phenomenon, we design an additional experiment through raising the tuning iteration to 120 and the training epoch to 400 on database CIFAR-100. The Figure 7 and Table 5 show the test accuracy and the recognition results. By
Table 4: Recognition results(%) on CIFAR-100 database. The results suggest that the CIPs gets relatively higher recognition rates and the BALD and Entropy methods have the nearly same results in the most of iterations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Train4</th>
<th>Train12</th>
<th>Train20</th>
<th>Train28</th>
<th>Train36</th>
<th>Train44</th>
<th>Train52</th>
<th>Train60</th>
<th>Train68</th>
<th>Train76</th>
<th>Train84</th>
<th>Train92</th>
<th>Train100</th>
</tr>
</thead>
<tbody>
<tr>
<td>VarRatio</td>
<td>5.49</td>
<td>12.04</td>
<td>13.46</td>
<td>19.03</td>
<td>21.34</td>
<td>25.64</td>
<td>27.60</td>
<td>29.29</td>
<td>31.30</td>
<td>32.98</td>
<td>28.72</td>
<td>34.43</td>
<td>35.15</td>
</tr>
<tr>
<td>Entropy</td>
<td>6.38</td>
<td>13.10</td>
<td>15.24</td>
<td>21.13</td>
<td>25.12</td>
<td>27.87</td>
<td>30.04</td>
<td>31.48</td>
<td>33.00</td>
<td>33.05</td>
<td>32.46</td>
<td>35.87</td>
<td>36.58</td>
</tr>
<tr>
<td>BALD</td>
<td>5.68</td>
<td>14.79</td>
<td>17.35</td>
<td>23.26</td>
<td>26.21</td>
<td>29.13</td>
<td>31.26</td>
<td>32.72</td>
<td>34.04</td>
<td>34.83</td>
<td>34.09</td>
<td>35.97</td>
<td>37.81</td>
</tr>
<tr>
<td>CIPs</td>
<td>7.78</td>
<td>17.17</td>
<td>24.38</td>
<td>28.81</td>
<td>31.27</td>
<td>32.41</td>
<td>35.01</td>
<td>36.37</td>
<td>37.82</td>
<td>39.16</td>
<td>38.66</td>
<td>39.82</td>
<td>40.18</td>
</tr>
</tbody>
</table>

Table 5: Recognition results(%) on CIFAR-100 database (running 120 tuning iterations with 400 epochs). In order to observe the results after the 100th iteration in detail, we sample every 4 iterations. Comparing to table 4, the results suggest that the recognition rates are raising when the tuning times and training epochs are increasing.

<table>
<thead>
<tr>
<th>Method</th>
<th>Train52</th>
<th>Train60</th>
<th>Train68</th>
<th>Train76</th>
<th>Train84</th>
<th>Train92</th>
<th>Train100</th>
<th>Train104</th>
<th>Train108</th>
<th>Train112</th>
<th>Train116</th>
<th>Train120</th>
</tr>
</thead>
<tbody>
<tr>
<td>VarRatio</td>
<td>26.88</td>
<td>29.59</td>
<td>30.81</td>
<td>31.28</td>
<td>32.57</td>
<td>34.20</td>
<td>35.91</td>
<td>35.84</td>
<td>37.33</td>
<td>36.02</td>
<td>37.83</td>
<td>38.00</td>
</tr>
<tr>
<td>Entropy</td>
<td>28.40</td>
<td>31.25</td>
<td>31.93</td>
<td>33.18</td>
<td>34.60</td>
<td>35.61</td>
<td>37.04</td>
<td>36.30</td>
<td>37.25</td>
<td>37.90</td>
<td>37.94</td>
<td>38.55</td>
</tr>
<tr>
<td>BALD</td>
<td>31.13</td>
<td>32.15</td>
<td>33.62</td>
<td>34.75</td>
<td>36.02</td>
<td>36.54</td>
<td>37.37</td>
<td>37.16</td>
<td>38.33</td>
<td>38.60</td>
<td>38.34</td>
<td>38.54</td>
</tr>
<tr>
<td>CIPs</td>
<td>36.14</td>
<td>37.88</td>
<td>38.84</td>
<td>39.84</td>
<td>40.29</td>
<td>40.90</td>
<td>41.69</td>
<td>42.01</td>
<td>41.91</td>
<td>42.33</td>
<td>41.83</td>
<td>42.67</td>
</tr>
</tbody>
</table>

Figure 7: The test accuracy comparison on CIFAR-100 database (running 120 tuning iterations with 400 epochs). Comparing with the results from figure 5(c), we can see that when the tuning iterations and epochs increase, the corresponding recognition accuracy enhances and the fluctuation decreases at the same time.

5 CONCLUSION AND FUTURE WORKS

Deep learning needs a huge number of labeled samples to get high recognition accuracy, which is not suitable for use in the most of real-world applications [33], where the labeled samples are scanty. For comparison, the human-beings usually select a few efficient materials to learn and get the ideal outcomes actively under the instruction from a teacher [15]. Inspired by human-beings’ learning process, we used the cognitive knowledge for deep learning mechanism through transforming the models’ cognitive errors to cognitive features, which were employed to select sensitive samples to boost the models’ performance gradually. The experiments demonstrated that the sample selecting method could choose the most effective samples aiming at the models’ recognition errors and our method got the better results than the baseline methods on different complexity objects. When the objects were more complex, the deep models needed more tuning iterations and training epochs to get higher recognition accuracy, which just likes the human-beings’ learning phenomenon, for example, the more difficult knowledge needs more times or endeavors to be mastered. We believe that our work might offer a line of thinking about the research of cognitive deep learning. Next, we will focus on the analysis of the human-beings’ cognition about the corresponding description method with related mathematics, and work on the cognitive evaluation criterion in deep learning applications.

ACKNOWLEDGMENTS

This work is supported in part by the Chinese State Scholarship Fund 201608370049, NSF IIS Award 1651902, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.