Nearest Neighbor Classifier Embedded Network for Active Learning

Fang Wan[1], **Tianning Yuan**[1], **Mengying Fu**[1], **Xiangyang Ji**[2], **Qingming Huang**[1] **Qixiang Ye**[1,*]
[1] University of Chinese Academy of Sciences, Beijing, China
[2] Tsinghua University, Beijing, China
{wanfang, qmhuang, qxye}@ucas.ac.cn, {yuantianning19, fumengying19}@mails.ucas.ac.cn, xyji@tsinghua.edu.cn

Abstract

Deep neural networks (DNNs) have been widely applied to active learning. Despite of its effectiveness, the generalization ability of the discriminative classifier (the softmax classifier) is questionable when there is a significant distribution bias between the labeled set and the unlabeled set. In this paper, we attempt to replace the softmax classifier in deep neural network with a nearest neighbor classifier, considering its progressive generalization ability within the unknown subspace. Our proposed active learning approach, termed nearest Neighbor Classifier Embedded network (NCE-Net), targets at reducing the risk of over-estimating unlabeled samples while improving the opportunity to query informative samples. NCE-Net is conceptually simple but surprisingly powerful, as justified from the perspective of the subset information, which defines a metric to quantify model generalization ability in active learning. Experimental results show that, with simple selection based on rejection or confusion confidence, NCE-Net improves state-of-the-arts on image classification and object detection tasks with significant margins.

Introduction

With the rise of deep neural networks (DNNs), image recognition has made unprecedented progress. Nevertheless, DNN models are typically trained on large-scale datasets which require intensive human effort for data annotation. Active learning, which interactively queries the data it wants to learn from, defines an effective method to reduce data annotation cost in practical image recognition applications (Settles 2012; Sener and Savarese 2018; Yoo and Kweon 2019; Sinha, Ebrahimi, and Darrell 2019).

Conventional active learning methods can be categorized into uncertainty-based methods (Settles 2012; Settles and Craven 2008; Luo, Schwing, and Urtasun 2013; Joshi, Porikli, and Papanikolopoulos 2009; Gal, Islam, and Ghahramani 2017) and representative-based methods (Guo 2010; Elhamifar et al. 2013; Yang et al. 2015; Sener and Savarese 2018). The objective is to find the most informative unlabeled samples based on their uncertainty or diversity. In DNN, the uncertainty/diversity is usually calculated upon the predictions of a softmax classifier, with the assumption that the feature and classifier trained on the labeled set can be always generalized to the unlabeled set.

In this study, we argue that such an assumption is problematic, particularly when there is a significant bias be-

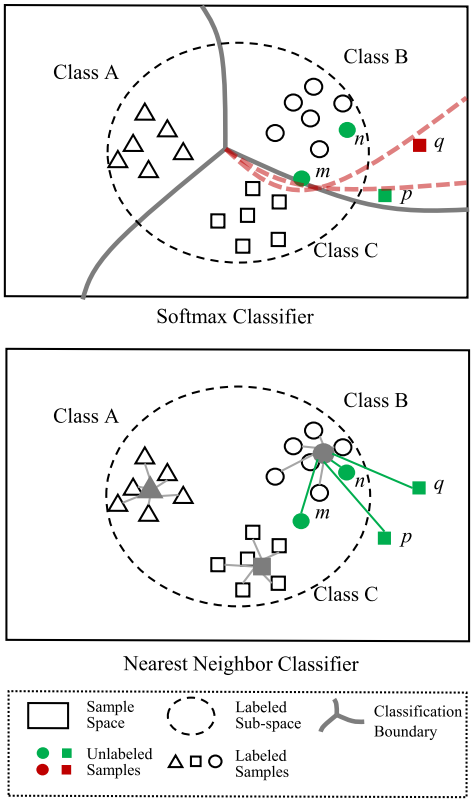


Figure 1: Comparison of the softmax classifier with the nearest neighbor classifier for sample selection. m and n are unlabeled samples within the labeled sub-space. p and q are unlabeled samples within the unlabeled space. The softmax classifier, when falsely generalized to the unlabeled space, tends to select sample m and p which are closer to the classification boundary but miss the informative sample q. The nearest neighbor classifier can select m, p and q, for its stronger generalization ability to the unknown space.

*Correspond author.

tween the distributions of labeled and unlabeled samples. We raise the concern through an intuitive analysis, Fig. 1, where the labeled samples cover a closed sub-space (within the dashed circle) while the unlabeled samples spread over an unknown space (outside the dashed circle). If the distributions of labeled and unlabeled samples are identical, the unlabeled samples close to the softmax classifier boundary (solid curves) are hard samples, which have large classification uncertainty and tend to be informative samples. However, if there is a significant bias between the distributions of labeled and unlabeled samples, the true classification boundary could be either of the dashed curves. The softmax classifier, a totally discriminative model based on assumption of the closed labeled set, tends to be falsely generalized to the unlabeled space and select sample m and p but misses the informative sample q.

To solve the false generalization problem, we propose to consider the simplest nearest neighbor classifier for active learning, based upon the common sense that simpler classifiers have stronger generalization ability. In Fig. 1(lower), when using a distance metric, the nearest neighbor classifier would assign significant score to m, p and q and make all of them selective. Once q is selected in an active learning cycle, the change of classification boundary (dashed curves) caused by the sample q is more significant than that caused by m and p, Fig. 1(upper). The essence behind the phenomenon is that the nearest neighbor classifier generalizes to its neighbors in a progressive fashion, avoiding making early decision on the unlabeled samples far away from the labeled ones.

Based upon above analysis, we propose to embed the nearest neighbor classifier to DNN for active learning. To guarantee the classification efficiency, we further propose to squeeze samples into prototype vectors, which predict the classification scores and adopt nearest neighbor classification embedding (NCE) loss using a non-linear activate function on distances. DNN driven by NCE loss, termed nearest neighbor classifier embedded network (NCE-Net), aims to reduce the risk of over-fitting labeled samples and facilitate selecting more informative unlabeled samples.

The contributions of this study include:

- We propose nearest neighbor classifier embedded network (NCE-Net), with the aim to improve generalization ability of models trained on the labeled sub-space upon the unlabeled sub-space in a simple-yet-effective way.
- We justify NCE-Net by sub-set information analysis, providing an effective metric to quantify model generalization ability in active learning.
- We applied NCE-Net to the image classification task, improving the state-of-the-arts of active learning with significant margins. We also extend NCE-Net to the object detection task, validating its task-agnostic advantage.

Related Works

Uncertainty-based Method. Active learning has been one of the most important research topic in machine learning and artificial intelligence areas for its practical application value. Conventional methods used uncertainty as a metric to select samples for active learning (Settles 2012). Uncertainty can be defined as the posterior probability of a predicted class (Lewis and Gale 1994; Lewis and Catlett 1994) or the margin between posterior probabilities of a predicted class and the secondly predicted class (Joshi, Porikli, and Papanikolopoulos 2009; Roth and Small 2006). It can be also calculated upon entropy (Settles and Craven 2008; Luo, Schwing, and Urtasun 2013; Joshi, Porikli, and Papanikolopoulos 2009), which is a natural metric about the uncertainty for probabilistic systems.

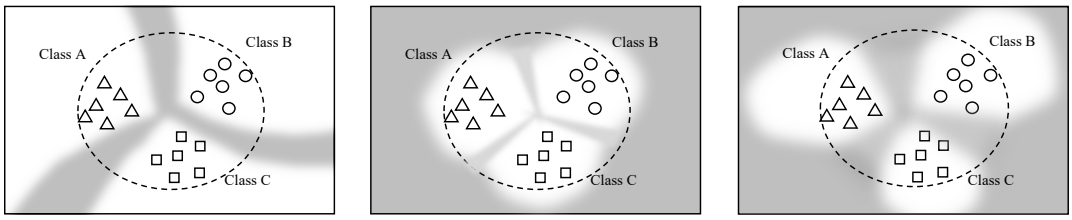
Uncertainty has been calculated using Monte Carlo Dropout and multiple forward passes (Gal, Islam, and Ghahramani 2017), with the aim to introduce Bayesian inference into the sample selection procedure. Despite of its effectiveness, the efficiency is significantly reduced for the usage of dense dropout layers which hinder the network convergence.

Representative-based Method. This line of methods estimate the distribution of unlabeled samples to find out representative samples. While discrete optimization methods (Guo 2010; Elhamifar et al. 2013; Yang et al. 2015) were employed to perform sample subset selection, the clustering method (Nguyen and Smeulders 2004) targeted at finding out the center points of subsets. The expected model change methods (Roy and McCallum 2001; Settles, Craven, and Ray 2007) utilized the present model to estimate expected gradient changes or expected output changes (Freytag, Rodner, and Denzler 2014; Käding et al. 2016), which guide the selection of informative samples.

A recent Core-set (Sener and Savarese 2018) method suggested that many active learning heuristics based on global distributions are not competent when applied to a batch of samples. It thus defined the active learning as a core-set selection problem, where a theoretical result is presented to characterize the performance of any selected subset using the geometry of the data points.

Learning Loss Method. In the deep learning era, many active learning methods remain falling into the uncertainty-based and representative-based routines (Lin et al. 2018; Wang et al. 2017; Beluch et al. 2018). Sophisticated methods have extended to open sets (Liu and Huang 2019), or combined it with self-paced learning (Tang and Huang 2019). Nevertheless, it remains questionable whether or not the intermediate feature representation is effective for sample selection. Recent learning loss approach (LL4AL) (Yoo and Kweon 2019) can be categorized either into an uncertainty approach or a distribution-based approach. By introducing the network structure to predict the "loss" of unlabeled samples, it can estimate sample uncertainty and diversity, and select samples with large "loss" in a fashion like hard negative mining.

Despite of the encouraging progress, the model generalization problem remains unsolved, which hinders the selection of the most informative samples. The context-aware methods (Hasan and Roy-Chowdhury 2015; Aodha et al. 2014) used the distance metric when selecting samples but remained exploring the limitation of discriminative classifiers. Recent adversarial classifiers (Sinha, Ebrahimi, and Darrell 2019; Zhang et al. 2020) and self-supervised learn-



(a) Entropy (Softmax) (b) Rejection (Ours) (c) Confusion (Ours)

Figure 2: Partitions of the unlabeled sub-space (outside the circle) using the models training within the labeled subspace (within the circle). Informative samples are located in the gray areas.

ing (Gudovskiy et al. 2020) provided interesting solutions for model generalization, but required higher computational and/or complexity cost. In this paper, we attempt solving the model generalization problem with a conceptually simple approach. Our work is inspired by the prototype learning method (Yang et al. 2018), but goes beyond it to handle unlabeled spaces.

Methodology

In this section, we first revisit active learning based on DNN with the softmax classifier. We then present the NCE-Net, which leverages a soft nearest neighbor classifier for sample selection under the guidance of "rejection" or "confusion" confidence. We finally justify NCE-Net by defining a subset information metric.

Active Learning with Softmax Classifier

The inputs of active learning consist of a small set of labeled images $\mathcal{X}_L^0$ with labels $\mathcal{Y}_L^0$, and a large set of unlabeled images $\mathcal{X}_U$. Active learning algorithms first train models on the labeled set $\mathcal{X}_L^0$. The trained model is then applied to select a subset $\mathcal{X}_L^1$ with M images from the unlabeled set $\mathcal{X}_U$ and query their labels $\mathcal{Y}_L^1$ for next cycle of training. M is usually far smaller than the total image number in $\mathcal{X}_U$.

The trained model consist of a backbone network and a softmax classifier. While the backbone network is for extracting features f_x for each image x, the softmax classifier, parameterized by w and b, computes the classification probability $p(f_x)$, as

$$p_c(f_x) = \frac{\exp(w_c \cdot f_x + b_c)}{\sum_c \exp(w_c \cdot f_x + b_c)}, \tag{1}$$

where w_c and b_c respectively denote the weight vector and bias for class c. The image classification loss is defined as

$$\mathcal{L}_{softmax} = -\log p_{y_x}(f_x), \tag{2}$$

where $y_x \in \{1, 2, ..., C\}$ denotes the label of images x and C the number of classes.

In each learning cycle i, sample selection targets at finding out a subset of samples, $\mathcal{X}_L^i$, which can boost the performance of the trained model to the largest extent in the next cycle. This procedure typically relies on the trained model and a sample selection metric such as entropy.

Soft Nearest Neighbor Classifier

The heuristics behind the active learning is that the softmax classifier trained on the labeled sub-space can be well generalized to the unlabeled sub-space for sample selection. Accordingly, the valuable samples will be found around the classification boundary (the gray region in Fig. 2(a)) while most samples in the unlabeled space (the white region outside the dashed cycle) are thought to have low uncertainty. However, as shown in Fig. 2(b), the unlabeled space (the gray region outside the dashed cycle) is actually an unknown area. When empirically generalizing the classification boundary of the softmax classifier to the whole unlabeled space, we face the risk of missing informative samples, which implies higher cost for sample annotation.

To solve this problem, we propose to replace the softmax classifier with a soft nearest neighbor classifier, which predicts the class label y^* of a test image x according to its soft distance to the labeled images, as

$$y^* = \arg\max_c \sum_i \delta(-d(f_x, f_{x_i}^c)), \tag{3}$$

where $f_{x_i}^c$ denotes the feature vector of image x_i with class label c. $d(\cdot)$ is an Euclidean or cosine distance function and $\delta(\cdot)$ is a non-linear activate function which projects the distance into [0, 1]. The distance $d(f_x, f_{x_i}^c)$ can be easily applied to evaluate how far the test image is from the labeled set, which can be used to calculate the uncertainty/value of sample for sample selection.

According to Eq. 3, labeled samples close to the test sample have large impact on its classification probability, while those far away from the test sample have little impact. In active learning, the nearest neighbor classifier preferentially predicts the unlabeled samples according to their distances to labeled ones, and avoids making early decision on the unlabeled samples far away from the labeled ones. For multiple learning cycles, it predicts their labels in a progressive generalization fashion, alleviating the false generalization caused by the softmax classifier.

Nearest Neighbor Classifier Embedded Network

According to Eq. 3, for each test image, it requires to compute the distances to all labeled images, which have the two following drawbacks: (1) It can not be used in the batch-mode DNNs; (2) It is computationally expensive, particularly when the dimensionality of features is high. To embed the nearest neighbor classifier to the deep neural networks for efficient classification, we propose to learn N prototype vectors $m_{c,n} (n = 1, ..., N$ for each class) using the labeled images. The classification likelihood based upon the proto-

type vectors is defined as

$$p_c(f_x) = \max_n \delta(-d(f_x, m_{c,n})), \tag{4}$$

where $m_{c,n}$ is the n-th prototype of class c, which is learned together with the feature extractor. During training, $m_{c,n}$ is randomly initialized and jointly updated by gradient descend with the network parameters.

By using the prototype vectors to substitute samples, the computational cost of the classifier is largely reduced. Meanwhile, the classification probability $p_c(f_x)$ can be inferred in a mini-batch mode. Accordingly, the loss function is defined using the binary cross entropy as

$$\mathcal{L}_{NCE} = -\sum_c y_{x,c} \log p_c(f_x) + (1 - y_{x,c}) \log(1 - p_c(f_x)), \tag{5}$$

where $y_{x,c} \in \{0, 1\}$.

In Eq. 4, $d(\cdot)$ is defined as either the Euclidean distance or the cosine distance. Take the Euclidean distance for example, the Gaussian kernel function is used for the activation function $\delta(\cdot)$. The likelihood of image x for class c is then defined as

$$p_c^{Euc}(f_x) = \max_n \exp\left(\frac{-||f_x - m_{c,n}||^2}{2R_c^2}\right), \tag{6}$$

where R_c is a learnable parameter indicating the rejection radius for class c. When the distance between feature f_x and the closest neighbor is larger than R_c, image x is categorized to class c, otherwise it is rejected by class c. Considering that the cosine distance is symmetric about the origin, we use Sigmoid as the activation function and define the likelihood as

$$p_c^{cos}(f_x) = \max_n \left(\frac{1}{1 + \exp\left(-cos(f_x, m_{c,n}) \cdot R_c\right)}\right). \tag{7}$$

Sample Selection

In each learning cycle, the trained model is applied to select unlabeled samples, according to either of the two following metrics based on the output of the proposed NCE-Net.

Rejection Confidence. To reflect the probability of an image being rejected by all of the image classes, the rejection confidence is defined as

$$\mathcal{M}_{rej}(x) = \sum_c 1 - p_c(f_x). \tag{8}$$

As illustrated in Fig. 2(b), the rejection confidence focuses on the sub-space (the gray area) covering the most samples of large uncertainty (the gray area between classes), where the samples are far from the labeled sub-space.

Confusion Confidence. To reflect how much the model is confused by the classes of non-maximum probability, the confusion confidence is defined as

$$\mathcal{M}_{conf}(x) = \sum_c \left(1 + p_c(f_x) - \max_c p_c(f_x)\right). \tag{9}$$

As illustrated in Fig. 2(c), the confusion confidence can be regarded as a special kind of entropy upon the output of the softmax classifier (Fig. 2(a)). While the most confused area is filled with samples with large classification uncertainty, it also includes samples far away from the labeled sub-space.

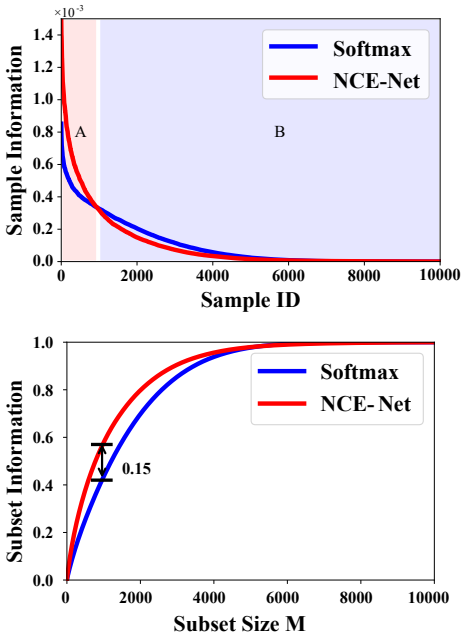


Figure 3: Comparison of subset information of selected samples by the proposed NCE-Net and DNN with the softmax classifier. The samples are sorted in the decreasing order according to their information (upper). The samples selected by NCE-Net have larger subset information (lower).

The images with the large rejection or confusion confidence are queried and added to the labeled set for the next training cycle. Note that with either the rejection or the confusion confidence, we select samples not only of large classification uncertainty but also far away from the labeled subspace. This facilities quickly filling the unlabeled space using the selected samples for efficient active learning.

Subset Information Analysis

We make an analysis on NCE-Net and DNN with the softmax classifier from the perspective of subset information, based on which the informative samples are selected in each active learning cycle. To quantify the information of sample, we define the event probability of a sample x_i being correctly classified as $p(x_i) = \max_c p_c(f_{x_i})$. According to the information theory (MacKay 2003), the information of an event can be defined as

$$I(x_i) = -\log p(x_i) = -\log(\max_c p_c(f_{x_i})), \tag{10}$$

which means that smaller event probability $p(x_i)$ brings more information. For an unlabeled set $\mathcal{X}_U$ with K samples, we suppose that its information is independent to classification models. We therefore set the information of $\mathcal{X}_U$ to 1 and define the information of sample x_i as

$$\hat{I}(x_i) = \frac{I(x_i)}{\sum_{x_i} I(x_i)}. \tag{11}$$

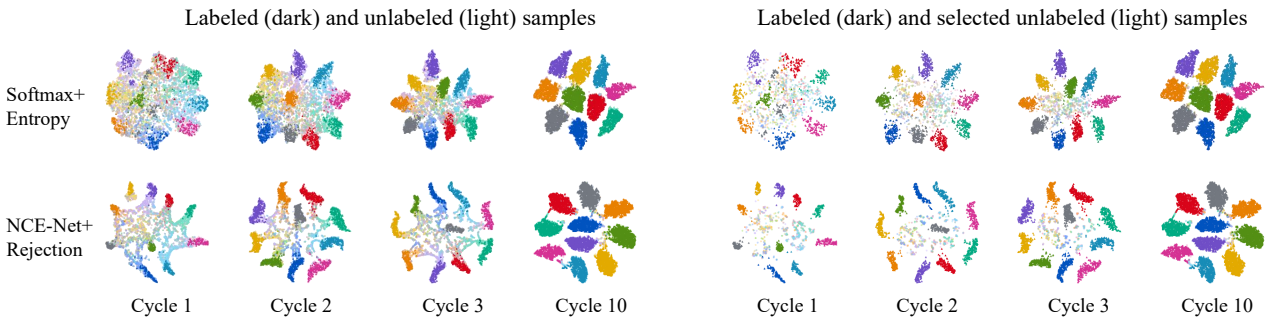


Figure 4: Visualization of sample distributions in different active learning cycles using t-SNE. (Best wiewed in color)

By sorting the information of $x_i \in \mathcal{X}_U$ in the decreasing order using 10000 randomly selected unlabeled images from CIFAR-10, we visualize the information of sample subsets in Fig. 3(upper). The model of the first learning cycle is used for both DNN with softmax and the NCE-Net. Sample information by NCE-Net is concentrated on less samples (area A in Fig. 3(upper)), which enables the learning model to benefit from selecting more informative samples with smaller query cost. The reason lies in that NCE-Net is able to reject the unknown images and output small probability $p(x_i)$ which indicating large information. Besides, it can classify the easy samples with large probability $p(x_i)$ to prevent the decentralization of information (area B in Fig. 3 (upper)). The ability of rejecting unknown samples and classifying labeled samples are essential for active learning methods.

When selecting a subset of M ($M << K$) unlabeled samples $\mathcal{X}_U^*$, the increase of information (subset information) is calculated as

$$\mathcal{I}(\mathcal{X}_U^*) = \sum_{x_i \in \mathcal{X}_U^*} \hat{I}(x_i). \tag{12}$$

In Fig. 3, it can be seen that when M is small, the subset information $\mathcal{I}_{NCE}(\mathcal{X}_U^*)$ is significantly larger than $\mathcal{I}_{softmax}(\mathcal{X}_U^*)$ (Fig. 3(lower)), justifying NCE-Net's effectiveness for informative sample selection.

Experiments

NCE-Net was evaluated on image classification and object detection tasks. In each task, we first introduced experimental settings and then reported the active learning performances. We then compared NCE-Net with the state-of-the-art methods.

Experimental Settings

For image classification, CIFAR-10 and CIFAR-100 were used as the benchmarks and the top-1 accuracy was used as the evaluation metric. All performances were averaged by three trials.

Datasets. CIFAR-10 dataset contains 60000 images of 10 object categories. Images were split into two sets where 50000 for $train$ and 10000 for $test$. The CIFAR-100 dataset is made up of 60000 images containing 100 object categories which are split into two subsets: 50000 for $train$, and 10000 for $test$. We applied random crop and random horizontal flip for data augmentation (Yoo and Kweon 2019). Images were normalized by the mean and standard deviation vectors of each channel estimated over the training set.

CNN Models. The ResNet-18 (He et al. 2016) was employed as the backbone network, following the settings in (Yoo and Kweon 2019; Zhang et al. 2020). To implement the NCE-Net, the last fully-connected (FC) layer of ResNet-18 was replaced with the soft nearest neighbor classifier.

Training Details. NCE-Net was implemented with Pytorch and run on a single NVIDIA RTX 2080Ti GPU. On CIFAR-10, we first initialized the labeled set $\mathcal{X}_L^0$ using 1000 randomly selected images. For each active learning cycle, we selected 1000 images from the unlabeled set and moved them into labeled set, until the labeled set increased to 10000. The classification model was trained for 200 epochs in each cycles, where the learning rate were set to 0.1 for the first 160 epoch and decreased to 0.01 for the remaining epochs. The batch size was set to 128. The momentum and the weight decay were set to 0.9 and 0.0005, respectively. For CIFAR-100, we initialized the labeled set $\mathcal{X}_L^0$ using 5000 randomly selected images and selected 2500 images in each cycle.

Model Effect

Visualization. In Fig. 4, we visualize the sample distributions from learning cycles. It can be seen that the unlabeled samples are clearly separated by NCE-Net, which benefits querying informative samples. The sample features extracted by the softmax classifier cover a large space for better generalization. However, this makes the labeled and unlabeled samples highly overlapped with each other and therefore aggregates the difficulty of querying informative samples.

In Fig. 4(right), we show the distributions of selected informative samples (light color). NCE-Net with rejection metric selects samples which are far from the labeled samples, while softmax classifier selects samples overlapped with the labeled samples. NCE-Net demonstrates progressive generalization ability within the unlabeled sub-space.

NCE-Net. In Table 1, NCE-Net was compared with the baseline method (network with the softmax classifier) under different sample selection metrics including "Random", "Entropy", "LL4AL"(Yoo and Kweon 2019), "Rejection" (Eq. 8) and "Confusion" (Eq. 9). "BEntropy" denotes firstly computing entropy for each class and then averaging them for the final selection score, solving the problem that the sum

Classifier	Method	Proportion of Labeled Samples										
		2%	4%	6%	8%	10%	12%	14%	16%	18%	20%	100%
Softmax	Random	51.20	61.72	69.85	76.09	79.64	83.20	84.56	85.79	86.61	87.28	92.63
	Entropy	51.20	61.49	70.62	77.83	81.81	85.75	86.94	88.62	89.60	90.39	
	LL4AL	51.20	64.68	75.36	81.09	83.54	86.91	**88.45**	**89.71**	90.29	90.56	
NCE-Net	Random	56.22	66.44	73.52	78.87	80.88	83.63	85.38	86.07	87.28	87.61	92.81
	BEntropy	56.22	60.12	70.73	77.84	81.79	84.92	87.09	88.01	89.19	90.21	
	LL4AL	**56.50**	67.87	76.7	81.54	83.94	86.35	87.76	89.07	89.73	90.41	
	Rejection-	56.22	67.83	76.05	80.82	83.78	86.22	87.68	88.87	89.78	90.54	
	Confusion-	56.22	67.64	75.85	80.63	83.26	86.32	87.74	89.16	90.19	90.73	
	Rejection	56.22	**69.38**	**77.32**	**82.33**	84.41	86.59	88.03	89.27	**90.21**	**91.01**	
	Confusion	56.22	69.17	76.91	82.16	**84.49**	**87.04**	87.98	89.43	90.14	90.72	

Table 1: Effect of sample selection metrics on CIFAR-10 using Resnet18 backbone.

N	Proportion of Labeled Samples						
	10%	15%	20%	25%	30%	35%	40%
1	37.60	48.19	55.33	60.36	63.25	66.25	68.36
2	**40.74**	**49.86**	**56.31**	**61.02**	**64.41**	66.86	**69.25**
3	39.43	49.16	56.22	60.92	63.58	**66.90**	68.77
5	38.19	48.81	55.77	60.07	63.45	66.12	68.60
10	33.90	48.07	55.21	60.15	63.16	66.16	68.19

Table 2: Evaluation of number N of nearest neighbors (prototypes) on CIFAR-100.

of output score of NCE-Net is not equal to 1. "LL4AL" is a state-of-the-art method, which predicts the learning loss of unlabeled sample (Yoo and Kweon 2019). Larger learning loss corresponds to more informative samples.

With the "Random" metric, NCE-Net achieved significantly better performance (56.22% vs. 51.20%) than the baseline when using 2% labeled images, and slightly better performance (87.61% vs. 87.28%) when using 20% labeled images. With the "Entropy" metric, NCE-Net improved the accuracy by 2.4% (90.21% vs. 87.61%) with 20% labeled images compared with the "Random" metric. It also outperformed the baseline with significant margins. When using the "LL4AL" method to select samples, NCE-Net achieved comparable performance with the baseline.

Sample Selection Metric. For "Rejection-" and "Confusion-", parameter R defined in Eq. 6 was fixed to 0.5. The proposed rejection and confusion metrics respectively outperformed the baseline by 2.93% and 3.12%, and outperformed the "BEntropy" by 0.33% and 0.52%. Note that under the same settings, "Rejection-" and "Confusion-" slightly outperformed the state-of-the-art method "LL4AL". With learnable R, "Rejection" outperformed the "Random" by 3.4% and "LL4AL" by 0.6%. As shown in Fig. 5, when using the cosine distance in NCE-Net (Eq.7), both "Rejection-Cos" and "Confusion-Cos" outperformed "BEntropy" and was on par with "LL4AL".

Number of Nearest Neighbors (N). In Table 2, we show the results of the number N of nearest neighbors (prototypes) on CIFAR-100. When $N = 1$, NCE-Net achieved 68.36% classification accuracy using 40% labeled images, which outperformed the state-of-the-art method (LL4AL) by 2.08%, demonstrating the effectiveness. The performance reached the best when $N = 2$. When N became larger, the performance slightly dropped. The reason could be that the CIFAR-100 dataset is a middle-scale dataset, upon which 2 nearest neighbors (prototypes) are sufficient to represent the labeled images. NCE-Net with larger N values might aggregate the risk of over-fitting problem but can achieve good performance given more training samples, $e.g.$, with 40% labeled images in the last column of Table 2.

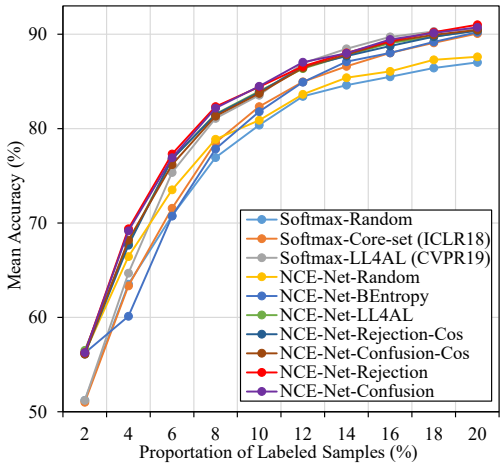


Figure 5: Comparison of NCE-Net with state-of-the-art methods including Core-set (Sener and Savarese 2018), VAAL (Sinha, Ebrahimi, and Darrell 2019), LL4AL (Yoo and Kweon 2019) and SRAAL (Zhang et al. 2020) on CIFAR-10 using ResNet-18.

.

Training Time. We evaluated the training time of NCE-Net and Softmax approaches on CIFAR10 using a NVIDIA GTX 1080Ti GPU. Softmax+Random costs 1.50 hours for 10 cycles of active learning, while NCE-Net+Rejection costs 1.51 hours and Softmax+LL4AL costs 1.55 hours. It is obvious that compared with the baseline method the training

Classifier	Method	Number of Labeled Samples										
		1k	2k	3k	4k	5k	6k	7k	8k	9k	10k	16.55k
Softmax	Core-set	**52.36**	62.34	65.87	67.67	68.79	69.44	70.16	70.83	71.16	71.72	77.43
	LL4AL	**52.36**	60.92	64.87	66.88	69.04	70.34	71.5	72.16	72.74	73.39	
NCE-Net	Random	52.28	59.75	63.84	67.46	68.86	70.60	71.31	72.45	73.45	73.97	77.59
	Rejection	52.28	**62.97**	**67.16**	**70.04**	**71.34**	**72.87**	**73.58**	**74.55**	**75.19**	**75.75**	

Table 3: Object detection performance and comparison with the state-of-the-art methods including Core-set and LL4AL which use softmax classifers on PASCAL VOC 2007.

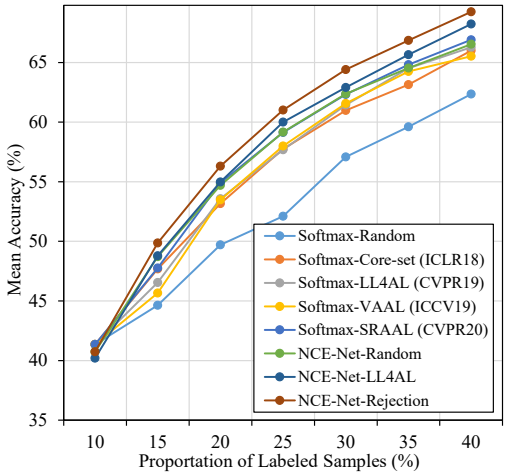


Figure 6: Comparison of NCE-Net with state-of-the-art methods on CIFAR-100 using ResNet-18.

and computational cost of our method is negligible. Furthermore, it runs faster than the SOTA method LL4AL.

Performance and Comparison on CIFAR-10. NCE-Net was compared with state-of-the-art approaches including Core-Set, VAAL, LL4AL, and SRAAL, Fig. 5. It respectively outperformed the compared methods which all used the softmax classifier, by 1.5% $\sim$ 5% with 2%, 4% and 6% labeled images, which shows the superiority of NCE-Net. In the last cycle, with 20% samples, NCE-Net achieved 91.01% accuracy, which was very close to (1.80% lower than) that on the full training set. With NCE-Net, the proposed rejection metric also outperformed the state-of-the-art LL4AL by 0.6% (91.01% vs. 90.41%).

Performance and Comparison on CIFAR-100. Fig. 6 shows that "NCE-Net-Rejection" significantly outperformed all other methods with softmax classifier. Particularly, it respectively outperformed the state-of-the-arts by 2.51%, 4.40%, and 2.25% when using 10%, 15% and 20% samples. We implemented LL4AL with NCE-Net to evaluate the sample selection metric. It can be seen that with a simple rejection metric, "NCE-Net-Rejection" outperformed "NCE-Net-LL4AL" in all cycles.

Object Detection

Dataset. For object detection, the PASCAL VOC 2007 and 2012 datasets (Mark et al. 2010) were used for evaluation. The VOC 2007 $trainval$ and VOC 2012 $trainval$ were merged as the training set, which contains 16551 images. The VOC 2007 $test$ was used to evaluate the mean Average Precision (mAP). All images are set to 300×300 with standard channel-wise normalization.

Base Detector. Following LL4AL (Yoo and Kweon 2019), SSD equipped with the VGG-16 backbone (Simonyan and Zisserman 2015) was employed as the base detector. In SSD, the last convolutional layer with kernel size $N \times (A \times C) \times 3 \times 3$ was used for bounding-box classification, where A, N and C are the number of anchors, channels and classes respectively. We replaced it with an $N \times (A \times N) \times 3 \times 3$ convolutional layer to extract features for each anchor and used the soft nearest neighbor classifier defined in Eq. 7 for classification.

Training Details. The labeled set $\mathcal{X}_L^0$ was initialized using 1000 randomly selected images. In each learning cycle, 1000 images was selected and added to the labeled set, until the labeled set increased to 10000. The detector was trained for 150 epochs in each cycle, where the learning rate was set to 0.001, and decreased to 0.0001 after 100 epochs and 0.00001 after 125 epochs. The batch size was set to 8. The momentum was set to 0.9 and the weight decay 0.0005.

Performance and Comparison. In Table 3, NCE-Net ("Random") achieved comparable results with the softmax-based SSD with 1000 initial images. When using the "Rejection" strategy to select samples, NCE-Net significantly outperformed the baseline detector in all cycles. When using 10000 images, NCE-Net outperforms "Random" by 1.78% (75.75% vs. 73.97%) and softmax classifier based "LL4AL" by 2.36% (75.75% vs. 73.39%).

Conclusion

In this paper, we proposed a conceptually simple method, termed nearest neighbor classifier embedded network (NCE-Net), for active learning. NCE-Net can reduce the risk of over-estimating the unlabeled samples and is less likely to miss the informative samples. NCE-Net was supported by intuitive analysis and justified from the perspective of the subset information analysis, which provides a way to quantify the ability of model generalization in active learning. Experiments on image classification and object detection benchmarks validate the superior performance and the task-agnostic advantage of NCE-Net. The proposed NCE-Net method provides a fresh insight to the classical active learning problem.

Acknowledgements

This work was supported in part by National National Natural Science Foundation of China (NSFC) under Grant 61836012, 61771447 and 62006216, Strategic Priority Research Program of Chinese Academy of Science under Grant XDA27010303, and Post Doctoral Innovative Talent Support Program of China under Grant 119103S304.

References

Aodha, O. M.; Campbell, N. D. F.; Kautz, J.; and Brostow, G. J. 2014. Hierarchical Subquery Evaluation for Active Learning on a Graph. In *IEEE CVPR*, 564–571.

Beluch, W. H.; Genewein, T.; Nürnberger, A.; and Köhler, J. M. 2018. The Power of Ensembles for Active Learning in Image Classification. In *IEEE CVPR*, 9368–9377.

Elhamifar, E.; Sapiro, G.; Yang, A. Y.; and Sastry, S. S. 2013. A Convex Optimization Framework for Active Learning. In *IEEE ICCV*, 209–216.

Freytag, A.; Rodner, E.; and Denzler, J. 2014. Selecting Influential Examples: Active Learning with Expected Model Output Changes. In Fleet, D. J.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *ECCV*, volume 8692, 562–577.

Gal, Y.; Islam, R.; and Ghahramani, Z. 2017. Deep Bayesian Active Learning with Image Data. In Precup, D.; and Teh, Y. W., eds., *ICML*, volume 70, 1183–1192.

Gudovskiy, D. A.; Hodgkinson, A.; Yamaguchi, T.; and Tsukizawa, S. 2020. Deep Active Learning for Biased Datasets via Fisher Kernel Self-Supervision. In *IEEE CVPR*, 9038–9046.

Guo, Y. 2010. Active Instance Sampling via Matrix Partition. In Lafferty, J. D.; Williams, C. K. I.; Shawe-Taylor, J.; Zemel, R. S.; and Culotta, A., eds., *NeurIPS*, 802–810.

Hasan, M.; and Roy-Chowdhury, A. K. 2015. Context Aware Active Learning of Activity Recognition Models. In *IEEE ICCV*, 4543–4551.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *IEEE CVPR*, 770–778.

Joshi, A. J.; Porikli, F.; and Papanikolopoulos, N. 2009. Multi-class active learning for image classification. In *IEEE CVPR*, 2372–2379.

Käding, C.; Rodner, E.; Freytag, A.; and Denzler, J. 2016. Active and Continuous Exploration with Deep Neural Networks and Expected Model Output Changes. *CoRR* abs/1612.06129.

Lewis, D. D.; and Catlett, J. 1994. Heterogeneous Uncertainty Sampling for Supervised Learning. In Cohen, W. W.; and Hirsh, H., eds., *Machine Learning*, 148–156.

Lewis, D. D.; and Gale, W. A. 1994. A Sequential Algorithm for Training Text Classifiers. In Croft, W. B.; and van Rijsbergen, C. J., eds., *SIGIR*, 3–12.

Lin, L.; Wang, K.; Meng, D.; Zuo, W.; and Zhang, L. 2018. Active Self-Paced Learning for Cost-Effective and Progressive Face Identification. *IEEE PAMI* 40(1): 7–19.

Liu, Z.; and Huang, S. 2019. Active Sampling for Open-Set Classification without Initial Annotation. In *AAAI*, 4416–4423.

Luo, W.; Schwing, A. G.; and Urtasun, R. 2013. Latent Structured Active Learning. In Burges, C. J. C.; Bottou, L.; Ghahramani, Z.; and Weinberger, K. Q., eds., *NeurIPS*, 728–736.

MacKay, D. J. C. 2003. *Information theory, inference, and learning algorithms*. Cambridge University Press.

Mark, E.; Luc, V. G.; KI, W. C.; John, W.; and Andrew, Z. 2010. The Pascal visual object classes (voc) challenge. *Int. J. Comput. Vis* 88(2): 303–338.

Nguyen, H. T.; and Smeulders, A. W. M. 2004. Active learning using pre-clustering. In Brodley, C. E., ed., *ICML*, volume 69.

Roth, D.; and Small, K. 2006. Margin-Based Active Learning for Structured Output Spaces. In Fürnkranz, J.; Scheffer, T.; and Spiliopoulou, M., eds., *ECML*, volume 4212, 413–424.

Roy, N.; and McCallum, A. 2001. Toward Optimal Active Learning through Sampling Estimation of Error Reduction. In Brodley, C. E.; and Danyluk, A. P., eds., *ICML*, 441–448.

Sener, O.; and Savarese, S. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *ICLR*.

Settles, B. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning.

Settles, B.; and Craven, M. 2008. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *EMNLP*, 1070–1079.

Settles, B.; Craven, M.; and Ray, S. 2007. Multiple-Instance Active Learning. In Platt, J. C.; Koller, D.; Singer, Y.; and Roweis, S. T., eds., *NeurIPS*, 1289–1296.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Bengio, Y.; and LeCun, Y., eds., *ICLR*.

Sinha, S.; Ebrahimi, S.; and Darrell, T. 2019. Variational Adversarial Active Learning. In *IEEE ICCV*, 5971–5980.

Tang, Y.; and Huang, S. 2019. Self-Paced Active Learning: Query the Right Thing at the Right Time. In *AAAI*, 5117–5124.

Wang, K.; Zhang, D.; Li, Y.; Zhang, R.; and Lin, L. 2017. Cost-Effective Active Learning for Deep Image Classification. *IEEE CSVT* 27(12): 2591–2600.

Yang, H.; Zhang, X.; Yin, F.; and Liu, C. 2018. Robust Classification With Convolutional Prototype Learning. In *IEEE CVPR*, 3474–3482.

Yang, Y.; Ma, Z.; Nie, F.; Chang, X.; and Hauptmann, A. G. 2015. Multi-Class Active Learning by Uncertainty Sampling with Diversity Maximization. *IJCV* 113(2): 113–127.

Yoo, D.; and Kweon, I. S. 2019. Learning Loss for Active Learning. In *IEEE CVPR*, 93–102.

Zhang, B.; Li, L.; Yang, S.; Wang, S.; Zha, Z.; and Huang, Q. 2020. State-Relabeling Adversarial Active Learning. In *IEEE CVPR*, 8753–8762.