

# Improving DCNN Performance with Sparse Category-Selective Objective Function

Shizhou Zhang, Yihong Gong\*, Jinjun Wang

Institute of Artificial Intelligence and Robotics,  
Xi'an Jiaotong University, Xi'an, 710049, P.R.China

## Abstract

In this paper, we choose to learn useful cues from object recognition mechanisms of the human visual cortex, and propose a DCNN performance improvement method without the need for increasing the network complexity. Inspired by the category-selective property of the neuron population in the IT layer of the human visual cortex, we enforce the neuron responses at the top DCNN layer to be category selective. To achieve this, we propose the Sparse Category-Selective Objective Function (SCSOF) to modulate the neuron outputs of the top DCNN layer. The proposed method is generic and can be applied to any DCNN models. As experimental results show, when applying the proposed method to the “Quick” model and NIN models, image classification performances are remarkably improved on four widely used benchmark datasets: CIFAR-10, CIFAR-100, MNIST and SVHN, which demonstrate the effectiveness of the presented method.

## 1 Introduction

In recent years, Deep Convolutional Neural Networks (DCNN) have shown state-of-the-art performances with many applications in computer vision [Min Lin, 2014; Girshick *et al.*, 2014; Goodfellow *et al.*, 2013; Ioffe and Szegedy, 2015; Krizhevsky *et al.*, 2012], speech recognition [Dahl *et al.*, 2012; Hannun *et al.*, 2014], natural language processing [Collobert and Weston, 2008; Mnih and Hinton, 2009], *etc.* The great success of DCNN models can be attributed to the following key factors: 1) developments of large-scale, deep models that can accurately model complex problems, and the availability of big training data with millions of labeled examples to train large-scale models, 2) the introduction of many training tricks, such as the rectifier activation function [Krizhevsky *et al.*, 2012], Dropout [Hinton *et al.*, 2012], DropConnect [Wan *et al.*, 2013], that can effectively prevent models’ co-adaptation and overfitting problems, and 3) high performance computing technologies and

platforms [Dean *et al.*, 2012] that allow large-scale models to be trained in affordable times.

Generally speaking, deeper and wider (more feature maps per convolution layer) models along with more training data lead to better performance accuracies [Simonyan and Zisserman, 2015; Szegedy *et al.*, 2015]. However, these strategies to improve DCNN performances are approaching their limitations. As elaborated in Section 2, when the model depth and complexity reach certain levels, further increasing the number and size of network layers will be less and less effective at improving the network performance. Moreover, very deep models usually do not converge when trained using the standard BP algorithm, and pre-training lower layers becomes a must [Simonyan and Zisserman, 2015]. Besides, training very deep models (*e.g.* GoogLeNet [Szegedy *et al.*, 2015], GoogLe-BN [Ioffe and Szegedy, 2015]) usually requires CPU/GPU clusters and ultra-large training data, and will be out of reach of small research groups with a limited research budget.

In this paper, we propose a DCNN performance improvement method that does not need increased network complexity. We choose to learn useful cues from the object recognition mechanisms of the human visual cortex. The human vision system outperforms existing machine vision systems at almost all tasks. Therefore, building a system that emulates certain properties of the human visual system has always been a promising research topic. In fact, CNN itself borrows the ideas of “local receptive fields”, “shared weights”, “spatial pooling”, *etc.*, from the properties of the primate visual cortex.

In recent decades, multi-disciplinary research efforts from neuroscience, physiology, psychology, *etc.*, have discovered that object recognition in the human visual cortex is modulated via the ventral stream [Gross, 1994; Miyashita, 1993], starting from the primary visual cortex (V1) through extrastriate visual areas I (V2) and IV (V4), to the inferotemporal cortex (IT), and then to the PreFrontal Cortex (PFC). Through this layered structure, raw neuronal signals from the retina are gradually transformed into higher level representations that are discriminative enough for accurate and speedy object recognition. Research studies have revealed a category-selective property of the neuron population in the IT layer. More specifically, although each neuron’s response in the IT layer is not unique to a specific class of objects, it responds

\*Corresponding author: Yihong Gong(ygong@mail.xjtu.edu.cn)

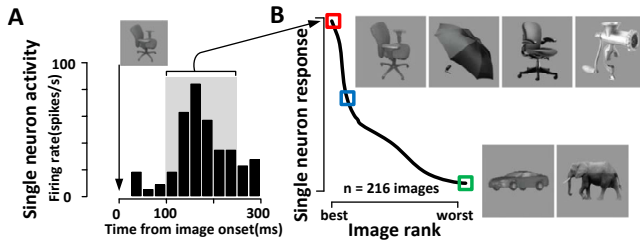


Figure 1: The category-selective property of one typical neuron in IT. (A) Poststimulus spike histogram from an example IT neuron to one object image (a chair) that was most effective among 216 tested object images. The responses in the gray time window are averaged. (B) The mean responses of the same IT neuron to each of 216 object images. As is typical, the neuron responded strongly to 10% of object images in some categories (four example images of nearly equal effectiveness are shown) and was suppressed below background rate by other objects belonging to other categories (two example images shown). This figure is from [DiCarlo *et al.*, 2012].

only to about 10% of all the object categories in the real world, and remains inactive in response to the rest [DiCarlo *et al.*, 2012]. Through this category-selective property, object identity of the input signal becomes available at the IT layer [DiCarlo *et al.*, 2012]. Figure 1 shows the category-selective property of a sample IT neuron.

Inspired by the above property of the IT neuron population, we propose to improve DCNN models by enforcing the neuron responses at the top DCNN layer to be category selective. To achieve this, we propose an enhancing cost function named the Sparse Category-Selective Objective Function (SCSOF) by using the  $L_{2,1}$  Norm to modulate the neuron outputs at the top DCNN layer. When the SCSOF is used together with other classification cost function, *e.g.* K-L divergence, the performance can be greatly improved. Our proposed DCNN performance improvement method is independent of any choice of DCNN model. As experimental results show, when applying the proposed method to the current state-of-the-art DCNN models, image classification performances are remarkably improved on several benchmark datasets. To summarize, the contributions of this paper are as follows:

- We propose to improve a DCNN model by enforcing the top-layer’s activation to be category-selective. Since no assumption is made to the DCNN architecture, the proposed principle borrowed from the ventral visual stream is generic and can be applied to different DCNN models;
- We propose the SCSOF by using the  $L_{2,1}$  norm to modulate the neuron outputs at the top DCNN layer. This approach makes an explicit analogy between the model’s top layer neuron responses and those of the IT neuronal population. The enhancing cost function implicitly propagates into the learned model parameters, such that the extracted features at the model’s top-layer present clear category-selectivity;
- Experimental results validate that the category-selective

property is achievable via the proposed enhancing cost function. Besides, experimental results on four benchmark datasets show superior performances over existing DCNN models.

The remainder of the paper is organized as follows: Section 2 reviews the related works. Section 3 presents the proposed DCNN performance improvement method. Experimental results are shown in Section 4. And we draw the conclusion in Section 5.

## 2 Related Works

Methods which improve the performance of DCNN models mainly include increasing the model complexity [Simonyan and Zisserman, 2015; Szegedy *et al.*, 2015], enlarging the training data [Krizhevsky *et al.*, 2012; Simonyan and Zisserman, 2015] and exploiting well-designed loss functions [Chopra *et al.*, 2005; Schroff *et al.*, 2015] and training techniques [Hinton *et al.*, 2012; Wan *et al.*, 2013]. In this section, we focus on reviewing more relevant works which improve the DCNN models when a specific network and the training data is given. Conventionally, the K-L divergence combined with the softmax activation function is adopted to measure the distance between the predicted label and the groundtruth. Contrastive loss function is well-designed for the so-called Siamese Network [Chopra *et al.*, 2005] which outputs one or zero for the intra-class pair and inter-class pair input, respectively. Further, the triplet-loss function minimizes the distance between the anchor point and the positive point and maximizes that between the anchor point and the negative point simultaneously, which is elaborated for triplet inputs [Schroff *et al.*, 2015]. Both the contrastive loss function and triplet-loss function are well-designed for the task of verification and can not be used for the more generic image classification task. While in this paper, we propose an enhancing objective function which can be used for the classification task when combined with a conventional cost function, *e.g.* the K-L divergence with the softmax activation.

Specifically, we borrow the category-selective property which means that an IT neuron’s response to visual input is very sparse with respect to categories (or groups). We simulate this category-selective property using the group-sparsity model. In fact, the sparsity and group sparsity prior has been widely used in feature selection and learning literature. [Zhao *et al.*, 2015] proposed to tackle the heterogeneous feature selection problem by using sparse group LASSO. A weight vector is derived to indicate the importance of the feature groups and feature groups with large weights are considered more relevant and thus are selected. [Dong *et al.*, 2011] proposed a new classification method called locality-constrained group sparse representation for the task of human gait recognition. Each probe video is classified by minimizing the weighted mix-norm-regularized reconstruction error with respect to the gallery videos. [Sun and Ponce, 2013] utilized a group sparse regularizer to jointly select and optimize a set of discriminative part detectors in a max-margin framework. [Gregor *et al.*, 2011] proposed a method for structured sparse coding and dictionary design via lateral inhibition. [Stevens *et al.*, 2013] described a learning algorithm

which efficiently learns a sparse ranking function by using a domination loss. While in this paper, to the best of our knowledge, we are the first to propose using  $L_{2,1}$  norm to modulate the top layer of a ConvNet. The next section elaborates the proposed Sparse Category-Selective Objective Function and how we introduce it into a DCNN model.

### 3 Category-Selectivity in DCNN

In order to enforce the category-selective property, we focus on modulating the top layer's neuron outputs during the training phase. Specifically, the Sparse Category-Selective Objective Function based on the  $L_{2,1}$  norm is combined with the standard objective function, as elaborated in the next subsection.

#### 3.1 Framework and Formulation

For clarity, firstly we introduce the objective function of a conventional DCNN model. Let  $S = \{(\mathbf{X}_i, y_i), i = 1, \dots, N\}$  denote the input training data set, where  $\mathbf{X}_i \in \mathbb{R}^n$  denotes the  $i^{th}$  input datum,  $y_i \in \{1, \dots, K\}$  denotes its ground-truth label, and  $K$  is the number of classes. The goal of DCNN training is to learn layers of weights that minimize the classification error from the output layer. In the following discussion, the bias terms are absorbed into the weight parameters for simplicity. A recursive function for an  $M$ -layer CNN model can be defined as follows:

$$\mathbf{Z}^{(0)} \equiv \mathbf{X} \quad (1)$$

$$\mathbf{Z}^{(m)} = \mathbf{W}^{(m)} * \mathbf{Z}^{(m-1)} \quad (2)$$

$$\mathbf{A}^{(m)} = f(\mathbf{Z}^{(m)}) \quad (3)$$

where  $m = 1, \dots, M$ ,  $M$  denotes the number of layers,  $\mathbf{W}^{(m)}$  the weights to be learned,  $\mathbf{Z}^{(m)}$  the feature maps generated at layer  $m$ ,  $f(\cdot)$  is an element-wise nonlinear activation function, *e.g.* sigmoid, tanh or ReLU transfer function, or a pooling function on  $\mathbf{Z}^{(m)}$ , and  $\mathbf{A}^{(m)}$  denotes the responses after nonlinear activation or pooling. The entire set of weights can be denoted as  $\mathbf{W} = [\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(M)}]$ .

The objective function of the conventional DCNN model can be written as

$$L(\mathbf{W}) = \sum_{i=1}^N \mathcal{L}(\mathbf{X}_i, y_i) + \lambda \sum_{m=1}^M \|\mathbf{W}^{(m)}\|^2 \quad (4)$$

where  $\mathcal{L}(\mathbf{X}_i, y_i)$  is the classification error for sample  $\mathbf{X}_i$ ,  $\|\mathbf{W}^{(m)}\|^2$  is the weight decay term on the  $m^{th}$  layer, and  $\lambda$  is a parameter to balance the classification error on the training data set and the model parameters.

Assume that we are enforcing the category-selective property on the  $m^{th}$  layer of the DCNN model. Denote the neuron outputs of the  $m^{th}$  layer on the entire training set by  $\mathbf{A}^m \in \mathbb{R}^{p \times N}$ , where  $p$  is the number of neurons in the layer,  $N$  is the number of training samples. Therefore each column  $i$  of  $\mathbf{A}^m$  represents the responses from all the neurons in the  $m^{th}$  layer to the  $i^{th}$  sample, and each row  $j$  represents the responses of one specific neuron  $j$  on the entire training set. The category-selective property is enforced as the Sparse Category-Selective Objective Function which is denoted as

$l(\mathbf{A}^m, \mathbf{y})$  where  $\mathbf{y}$  contains all labels of the training dataset. Thus the complete cost function turns into

$$L(\mathbf{W}) = \sum_{i=1}^N \mathcal{L}(\mathbf{X}_i, y_i) + \lambda \sum_{m=1}^M \|\mathbf{W}^{(m)}\|^2 + \beta l(\mathbf{A}^m, \mathbf{y}) \quad (5)$$

where  $\lambda$  and  $\beta$  adjust the weight decay term and the Sparse Category-Selective Objective Function, respectively. In the next subsection, we elaborate on the Sparse Category-Selective Objective Function  $l(\mathbf{A}^m, \mathbf{y})$ .

#### 3.2 Sparse Category-Selective Objective Function

As Figure 1 depicts, a single IT neuron has the category-selective property such that it only responds strongly to about 10% of object categories and is suppressed below the background rate by the others [DiCarlo *et al.*, 2012]. In other words, an IT neuron's response to visual inputs is very sparse with respect to categories (or groups). In practice, the  $L_{2,1}$  norm is the most popular mathematical tool for enforcing the group sparsity. Therefore, we define the third term in Eqn.(5) as follows:

$$l(\mathbf{A}^m, \mathbf{y}) = \sum_{i=1}^p \sum_{k=1}^K \sqrt{\sum_{j=1, j \in \omega_k}^N a_{ij}^2}, \quad (6)$$

where  $a_{ij}$  denotes the element  $(i, j)$  of  $\mathbf{A}^m$ ,  $\omega_k$  represents training samples belonging to the  $k^{th}$  category, and  $p, K, N$  have the same meanings as in Section 3.1. Note that each row  $j$  of  $\mathbf{A}^m$  represents the responses of one specific neuron  $j$  on the entire training set. For each neuron  $j$ , we first calculate the  $L_2$  norm of its response values for all the training samples belonging to category  $\omega_k$ . Then the  $L_2$  norm values across all the categories  $\omega_1, \dots, \omega_K$  of the training set are added up together to achieve the  $L_{2,1}$  norm computation.

Note that  $\mathbf{A}^m$  depends on  $\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(m)}$  and is independent of  $\mathbf{W}^{(m+1)}, \dots, \mathbf{W}^{(M)}$ . Hence directly regularizing  $\mathbf{A}^m$  will affect the weights from layer 1 to  $m$  via gradient backpropagation. Since the IT area is near the final stage of the visual processing stream, we enforce the category selective property on the top layer of DCNN. In our experiments,  $m$  is taken as  $M$  or  $M - 1$ . The next subsection elaborates how to optimize Eqn.(5).

#### 3.3 Implementation Details

The Back-propagation algorithm is adopted to calculate the gradient with respect to the weights of the DCNN model. In Eqn.(5), the gradient of the classification error term and the weight decay term is straightforward, and here we focus on obtaining the derivation of the third term, the gradient of the Sparse Category-Selective Objective Function  $l(\mathbf{A}^m, \mathbf{y})$ .

For clarity, we neglect the superscription  $m$  in  $\mathbf{A}^m$ . Denote  $\mathbf{A}^m$  as  $\mathbf{A}$  and let  $\mathbf{A} = [\vec{\mathbf{a}}_1, \dots, \vec{\mathbf{a}}_N]$ . For the  $i^{th}$  row of  $\mathbf{A}$ , we calculate the  $L_2$  norm of samples belonging to each category and denote the  $K$  dimensional vector as  $\vec{\mathbf{b}}_i$

$$\vec{\mathbf{b}}_i = [b_{i1}, b_{i2}, \dots, b_{iK}] \quad (7)$$

where

$$b_{ik} = \sqrt{\sum_{j=1, j \in \omega_k}^N a_{ij}^2} \quad (8)$$

for  $k = [1, \dots, K]$  and  $i = [1, \dots, p]$ .

Let  $\vec{\mathbf{I}}_j = [0, \dots, 1, \dots, 0]$  denote the vectorized label of the  $j^{th}$  sample, which is encoded by the ‘‘one of c’’ coding method. The gradient of the Sparse Category-Selective Objective Function with respect to the  $j^{th}$  column of  $\mathbf{A}$  is calculated by

$$\frac{\partial l(\mathbf{A}^m, \mathbf{y})}{\partial \vec{\mathbf{a}}_j} = \text{diag}\left(\frac{1}{\vec{\mathbf{I}}_j \cdot \vec{\mathbf{b}}_1 + \epsilon}, \dots, \frac{1}{\vec{\mathbf{I}}_j \cdot \vec{\mathbf{b}}_p + \epsilon}\right) \times \vec{\mathbf{a}}_j \quad (9)$$

where  $\epsilon$  denotes a very small positive number which prevents the divider from being 0. According to Eqn.(9), the Sparse Category-Selective Objective Function information can be easily propagated to previous layers based on the Back-propagation algorithm.

In practice, we calculate the gradients on the mini-batch training samples in order to be compatible with the stochastic gradient descent method. In the next section, we show that by using the proposed SCSOF, the category selective property of the neuron responses is more apparent on the test set. More importantly, the recognition performance can be remarkably improved. The implementation is based on the Caffe [Jia *et al.*, 2014] package.

## 4 Experimental Evaluations

### 4.1 Overview

To evaluate the effectiveness of the proposed SCSOF for improving object recognition performances of CNN models, we conduct experimental evaluations using shallow and deep models, respectively. As for shallow model, we choose the named ‘‘Quick’’ model<sup>1</sup> provided by the official Caffe [Jia *et al.*, 2014] package and evaluate it on three benchmark datasets, namely CIFAR-10, CIFAR-100 and SVHN. While as for deep model, the well-known ‘‘Network In Network’’(NIN) [Min Lin, 2014] model is adopted and four benchmark datasets, namely CIFAR-10, CIFAR-100, SVHN and MNIST, are tested, the same as that in [Min Lin, 2014]. Note that we did not test the ‘‘Quick’’ model on the MNIST dataset because the ‘‘Quick’’ model requires  $32 \times 32$  color input samples while the samples on MNIST are  $28 \times 28$  gray images. For ‘‘Quick’’ model, the weight decay coefficient  $\lambda$  is set to 0.004, the momentum is set to 0.9. The initial learning rate is set to 0.001 and decreased by a factor of 10 for every 8,000 iterations. The training process is finished over 30,000 iterations. For ‘‘NIN’’ model, we strictly follow the settings as in [Min Lin, 2014] for each dataset. The only hyper-parameter  $\beta$  introduced by the proposed SCSOF is empirically set to  $[10^{-6}, 10^{-4}]$ .

<sup>1</sup>The model is available in the Caffe package.

### 4.2 Datasets

We evaluate the proposed SCSOF on four widely used benchmark datasets, namely CIFAR-10, CIFAR-100, MNIST and SVHN. The reason for choosing these datasets is because they contain a large amount of small images (about  $32 \times 32$  pixels), so that models can be trained by using computers with moderate configurations within reasonable time frames. Because of this, the four datasets have become very popular choices for deep network performance evaluations in the computer vision and pattern recognition research communities. Firstly, we describe the datasets in this subsection and then report the results in the following subsections.

**CIFAR-10 Dataset.** The CIFAR-10 dataset is composed of 10 classes of natural images, 50,000 for training and 10,000 for testing. Each image is  $32 \times 32$  in size and in RGB format.

**CIFAR-100 Dataset.** The CIFAR-100 dataset has the same image size and format as the CIFAR-10 dataset, but it contains 100 classes. The number of images in each class is one tenth of those in the CIFAR-10 dataset. Also, this dataset contains 50,000 images for training and 10,000 images for testing.

**MNIST Dataset.** The MNIST dataset consists of 0-9 hand written digits which are  $28 \times 28$  gray images. There are 60,000 training samples and 10,000 testing samples in total.

**SVHN Dataset.** The Street View House Numbers(SVHN) dataset, obtained by extracting house numbers from the Google Street View images, is composed of 630,420  $32 \times 32$  color images, including the training set, testing set and an extra set. Multiple digits may exist in the same image, and the task of this dataset is to recognize the digit located at the center of each image. It is similar in flavor to MNIST but incorporates an order of magnitude more labeled data (over 600,000 digit images) and comes from a significantly harder, unsolved, real world problem, i.e. recognizing digits and numbers in natural scene images.

### 4.3 Experiments with shallow model

In this subsection, the named ‘‘Quick’’ model provided by the official Caffe [Jia *et al.*, 2014] package is selected as the baseline model. This model consists of three convolution layers and one fully connected layer. The baseline model adopts the widely used classification cost function, i.e. softmax activation plus KL-divergence cost function. Our method additionally enforces the proposed SCSOF on the last fully connected layer. We compare the baseline model and the proposed method on three datasets, namely CIFAR-10, CIFAR-100, and SVHN dataset. Note that the MNIST dataset is not tested as the inputs of this dataset are  $28 \times 28$  gray images while the ‘‘Quick’’ model requires inputs of  $32 \times 32$  color images.

Experimental results of test set error rates on the three datasets are shown in Table 1. From Table 1, it can be observed that, compared with the baseline model, the proposed SCSOF can remarkably reduce the test set error rates by 5.17% on CIFAR-10 dataset, 3.52% on CIFAR-100 dataset and 3.59% on SVHN dataset, respectively. The performance improvements demonstrate the effectiveness of the proposed SCSOF.

Table 1: Test set top-1 error rate of the proposed method on Quick model.

Cost Function	CIFAR10	CIFAR100	SVHN
KL-divergence	23.47	55.87	8.92
KL-divergence+SCSOF	<b>18.3</b>	<b>52.35</b>	<b>5.33</b>
gains	5.17	3.52	3.59

Table 2: Test top-1 error rate on CIFAR-10 dataset.

Algorithm	Error rates
Stochastic Pooling [Zeiler and Fergus, 2013]	15.13
Maxout Networks [Goodfellow <i>et al.</i> , 2013]	11.68
P. Maxout [Springenberg and Riedmiller, 2014]	11.35
NIN [Min Lin, 2014]	10.41
DSN [Lee <i>et al.</i> , 2015]	9.78
Our method	<b>9.52</b>

#### 4.4 Experiments with Deep model

In this subsection, we evaluate the SCSOF on the well-known “NIN” models [Min Lin, 2014]. NIN model consists of 9 convolution layers and no fully connected layer. Indeed, it is a very deep model, with 6 more convolution layers than that of the “Quick” model. Similarly, the baseline NIN model adopts softmax activation plus KL-divergence cost function and our method additionally enforces the proposed SCSOF on the last layer. Four widely used benchmark datasets, including CIFAR-10, CIFAR-100, MNIST and SVHN, are used to compare the baseline NIN model and the proposed method.

For fair comparison, we strictly follow the training and testing protocols in [Min Lin, 2014]. The CIFAR-10 and CIFAR-100 datasets are preprocessed by the global contrast normalization and ZCA whitening as in [Min Lin, 2014], no data whitening was applied for the MNIST dataset. For SVHN, 400 samples per class selected from the training set and 200 samples per class from the extra set were used for validation, while the remaining 598,388 images of the training and the extra sets were used for training, which is also the same with that in [Min Lin, 2014; Goodfellow *et al.*, 2013; Lee *et al.*, 2015]. The validation set was only used for tuning hyper-parameters and was not used for training the model. Preprocessing of the dataset again follows [Min Lin, 2014; Goodfellow *et al.*, 2013; Lee *et al.*, 2015].

The evaluation results are shown in Table 2, 3, 4, 5, in terms of test set top-1 error rate on the four benchmark datasets, respectively. It can be seen from these tables that, the proposed method outperforms the baseline NIN model by 0.9% on the CIFAR-10 dataset, 1.7% on the CIFAR-100 dataset, 0.5% on the SVHN dataset and 0.17% on the MNIST dataset. The improvement on MNIST in terms of absolute percentage is not very large, because the baseline NIN model already achieves test error rate of 0.47%, which is almost a saturate accuracy on this dataset. However, in terms of relative reductions of test error rates, the number reached 36%, which is quite remarkable.

In Table 2, 3, 4, 5, we also include the evaluation results of some representative methods, including Stochastic Pooling [Zeiler and Fergus, 2013], Maxout Networks [Good-

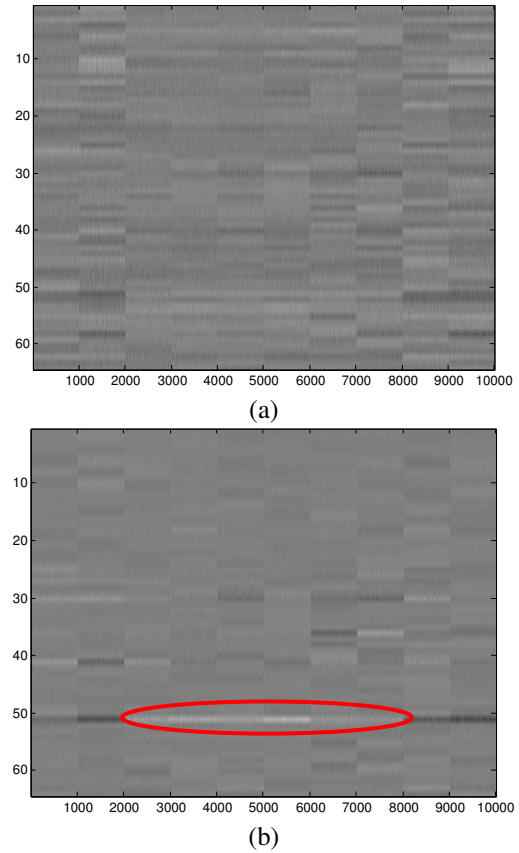


Figure 2: Visualization of neuron outputs. Each column  $i$  corresponds to all the neural responses to test image  $i$ , and each row  $j$  corresponds to the responses of neuron  $j$  to all the images in the test set. Images in the test set are re-ordered by group from category 1 to 10. (a) Neuron outputs without SCSOF enforced. (b) Neuron outputs with SCSOF enforced on the last fully connected layer. The variance of each neuron’s responses on a specific category in (a) is obviously larger than that in (b). Due to the effect of SCSOF, the category selectivity is more apparent in (b). Best viewed in electronic form.

fellow *et al.*, 2013], Probability Maxout [Springenberg and Riedmiller, 2014], Tree based priors [Srivastava and Salakhutdinov, 2013], Multi-digit Recognition [Goodfellow *et al.*, 2014], DropConnect [Wan *et al.*, 2013] and DSN [Lee *et al.*, 2015]. Additionally the proposed method outperforms the state-of-the-art DSN [Lee *et al.*, 2015] and it is worth mentioning that DSN is also based on the NIN structure with layer-wise supervision. The evaluation results can be summarized from the following three aspects: 1) The proposed SCSOF is model-independent and can be enforced on any CNN models. 2) When combined with shallow CNN model, the performance improvement is more remarkable. 3) Even SCSOF is enforced on state-of-the-art CNN models, the performance can be noticeably improved further.



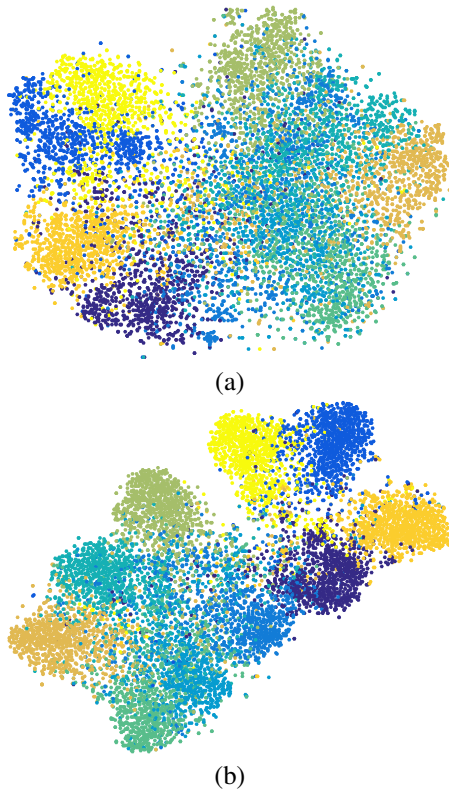


Figure 3: Feature visualization of the test dataset through t-SNE. One point represents one sample and points with the same color belong to the same class. (a) Features from the “Quick” model on CIFAR-10 test set without SCSOF. (b) Features from the “Quick” model with the SCSOF on the last fully connected layer. Obviously, features learned with SCSOF are more discriminant. Best viewed in electronic form.

#### 4.5 Visualization and Discussion

To gain some insights about what the proposed SCSOF does, we visualize the top layer feature representations on which the SCSOF is enforced. The last fully connected layer of the “Quick” model on CIFAR-10 test set, with or without the SCSOF enforced on, is selected to do the visualization.

At first, we reorder the test images category by category, namely the first 1000 images are “airplanes”, the second 1000 images are “automobiles”, the subsequent 1000 images are “birds”, ..., *etc.* In Figure 2, each row represents the activation response of one neuron. As Figure 2 shows, the category selective property is obviously more apparent if the proposed SCSOF is enforced. Interestingly, one neuron in Figure 2 (b), highlighted by a red ellipse, responds strongly to birds, cats, deers, dogs, frogs, and horses, i.e. those animal classes, and gets suppressed on non-animal classes. This shows that the neuron learns some high level semantic information of animals due to the Sparse Category-Selective Objective Function.

Next we use the t-SNE [Van der Maaten and Hinton, 2008] method to reduce the 64-D feature representations of the last

Table 3: Test top-1 error rate on CIFAR-100 dataset.

Algorithm	Error rates
Learned Pooling [Malinowski and Fritz, 2013]	43.71
Stochastic Pooling [Zeiler and Fergus, 2013]	42.51
Maxout Networks [Goodfellow <i>et al.</i> , 2013]	38.57
P. Maxout [Springenberg and Riedmiller, 2014]	38.14
Tree priors [Srivastava and Salakhutdinov, 2013]	36.85
NIN [Min Lin, 2014]	35.68
DSN [Lee <i>et al.</i> , 2015]	34.57
Our method	<b>34.03</b>

Table 4: Test top-1 error rate on MNIST dataset.

Algorithm	Error rates
Stochastic Pooling [Zeiler and Fergus, 2013]	0.47
Maxout Networks [Goodfellow <i>et al.</i> , 2013]	0.45
NIN [Min Lin, 2014]	0.47
DSN [Lee <i>et al.</i> , 2015]	0.39
Our method	<b>0.30</b>

Table 5: Test top-1 error rate on SVHN dataset.

Algorithm	Error rates
Stochastic Pooling [Zeiler and Fergus, 2013]	2.80
Maxout Networks [Goodfellow <i>et al.</i> , 2013]	2.47
P. Maxout [Springenberg and Riedmiller, 2014]	2.39
Multi Recognition [Goodfellow <i>et al.</i> , 2014]	2.16
DropConnect [Wan <i>et al.</i> , 2013]	1.94
NIN [Min Lin, 2014]	2.35
DSN [Lee <i>et al.</i> , 2015]	1.92
Our method	<b>1.90</b>

fully connected layer to 2-D and visualize the feature distribution in the 2-D space. It can be clearly seen from Figure 3 that features learned with the proposed SCSOF are more discriminant compared with the baseline method.

## 5 Conclusion

In this paper, inspired by the category-selective property of IT neuron population, we propose to improve DCNN models by enforcing the neuron responses at the top DCNN layer to be category selective. Specifically, we propose the Sparse Category-Selective Objective Function by using  $L_{2,1}$  Norm to modulate the outputs at the top DCNN layer. The proposed SCSOF is independent of any DCNN models and when it is combined with shallower CNN model, the performance improvement is very remarkable and even when it is enforced on state-of-the-art DCNN models, the performance can be further improved noticeably.

## Acknowledgments

This work is supported by National Basic Research Program of China (973 Program) under Grant No. 2015CB351705, and the National Natural Science Foundation of China (NSFC) under Grant No. 61332018.

## References

- [Chopra *et al.*, 2005] S. Chopra, R. Hadsell, and Y. Lecun. Learning a similarity metric discriminatively, with application to face verification. *CVPR*, 2005.
- [Collobert and Weston, 2008] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008.
- [Dahl *et al.*, 2012] G. E Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing.*, 20(1):30–42, 2012.
- [Dean *et al.*, 2012] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, and Q. V Le. Large scale distributed deep networks. In *NIPS*, 2012.
- [DiCarlo *et al.*, 2012] J. J DiCarlo, D. Zoccolan, and N. C Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434, 2012.
- [Dong *et al.*, 2011] Xu Dong, Huang Y., Zeng Z., and Xu X. Human gait recognition using patch distribution feature and locality-constrained group sparse representation. *IEEE Transactions on Image Processing*, 21(1):316–26, 2011.
- [Girshick *et al.*, 2014] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.
- [Goodfellow *et al.*, 2013] I. J Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *ICML*. 2013.
- [Goodfellow *et al.*, 2014] I. J Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. In *ICLR*. 2014.
- [Gregor *et al.*, 2011] K. Gregor, A. Szlam, and Y. Lecun. Structured sparse coding via lateral inhibition. *NIPS*, 2011.
- [Gross, 1994] C. G Gross. How inferior temporal cortex became a visual area. *Cerebral cortex*, 4(5):455–469, 1994.
- [Hannun *et al.*, 2014] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, and A. Coates. Deepspeech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.
- [Hinton *et al.*, 2012] G. E Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [Ioffe and Szegedy, 2015] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [Jia *et al.*, 2014] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014.
- [Krizhevsky *et al.*, 2012] A. Krizhevsky, I. Sutskever, and G. E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [Lee *et al.*, 2015] C. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AISTATS*. 2015.
- [Malinowski and Fritz, 2013] M. Malinowski and M. Fritz. Learnable pooling regions for image classification. In *ICLR workshop*. 2013.
- [Min Lin, 2014] S. Yan Min Lin, Q. Chen. Network in network. In *ICLR*. 2014.
- [Miyashita, 1993] Y. Miyashita. Inferior temporal cortex: where visual perception meets memory. *Annual review of neuroscience*, 16(1):245–263, 1993.
- [Mnih and Hinton, 2009] A. Mnih and G. E Hinton. A scalable hierarchical distributed language model. In *NIPS*, 2009.
- [Schroff *et al.*, 2015] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.
- [Simonyan and Zisserman, 2015] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*. 2015.
- [Springenberg and Riedmiller, 2014] J.T. Springenberg and M. Riedmiller. Improving deep neural networks with probabilistic maxout units. In *ICLR Workshop Track*. 2014.
- [Srivastava and Salakhutdinov, 2013] N. Srivastava and R. R Salakhutdinov. Discriminative transfer learning with tree-based priors. In *NIPS*, 2013.
- [Stevens *et al.*, 2013] M. Stevens, S. Bengio, and Y. Singer. *Efficient Learning of Sparse Ranking Functions*. Springer, 2013.
- [Sun and Ponce, 2013] J. Sun and J. Ponce. Learning discriminative part detectors for image classification and cosegmentation. In *ICCV*, 2013.
- [Szegedy *et al.*, 2015] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*. 2015.
- [Van der Maaten and Hinton, 2008] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.
- [Wan *et al.*, 2013] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus. Regularization of neural networks using dropconnect. In *ICML*, 2013.
- [Zeiler and Fergus, 2013] M. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. In *ICLR*. 2013.
- [Zhao *et al.*, 2015] L. Zhao, Q. Hu, and W. Wang. Heterogeneous feature selection with multi-modal deep neural networks and sparse group lasso. *IEEE Transactions on Multimedia.*, 17, 2015.