CS365: Image Classification Using Self-taught Learning For Feature Discovery

Harshvardhan Sharma, Nikunj Agrawal
Advisor: Prof. Amitabha Mukerjee

Indian Institute of Technology, Kanpur

April 24, 2014

Abstract

Image classification is an important task in computer vision which aims at classifying images based on their content. Most techniques for this task require a lot of labeled data to train the model which is scarce and expensive. Self-taught learning[2] tackles this problem by using a generic dataset of unlabeled data to train a generative model and use it for feature discovery. Deep learning techniques have been known to perform well for feature discovery. Convolutional deep belief networks[1] with probabilistic max pooling provide a translational invariant hierarchical generative model supporting both top-down and bottom-up inference that perform well on images. In this work, we employ CDBNs for self-taught learning to learn features from images to classify them.

1 Introduction

One of the important objectives of computer vision is automatic identification of objects in an image. This is achieved by learning from a dataset consisting of images from all categories and classifying the given image into one of these categories. However, this approach has a serious limitation, the unavailability of a labeled dataset of images from all categories. Labeled data is not easily available making it a very tedious process to build such datasets. Techniques like semi-supervised learning and transfer learning address this problem by using unlabeled data but these techniques still require the data from a similar domain. Self-taught learning however makes use of very generic unlabeled data which need not have the same generative distribution as the labeled data.

2 Previous work

In 2007, Raina et al[2] used it to match many state of the art results in different domains like image classification, music genre classification and UseNet article classification. In 2009, Lee et al. [1] used CDBNs to match the best results on Caltech101 dataset using a model which was very generic in nature and used limited training examples. The state of the art results for Caltech 101 is 67% overall, averaged on all the classes. In ImageNet LSVRC-2010 contest, Krizhevsky et al. [3] trained a convolutional deep belief network on 1.2 million images to classify them into 1000 different classes. The results proved to be considerably better than the previous state-of-the-art results.

3 Preliminaries

3.1 Self-taught learning

Self Taught learning uses sparse coding to learn higher level features from unlabeled data. The model which was trained using unlabeled images is then used to extract features for the labeled data. These additional features are augmented with the input image to get the new labeled data. Then this labeled data is used to train a supervised machine learning model, which yields significantly improved performance in the classification task.

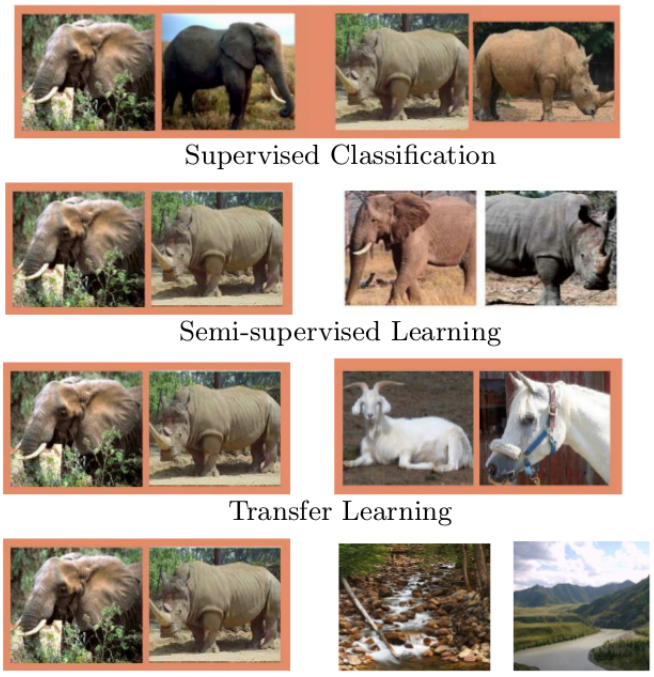

Figure 1: Comparison of machine learning methods. Images with orange boundaries are labeled and rest unlabeled. Source: Raina et al, ICML-07

The picture shown in Figure 1 shows the difference between different paradigms of machine learning used for classification. Here, the task is to classify images of the target dataset containing elephants and rhinos.

- Supervised learning requires labeled training and test sets of elephants and rhinos.
- Semi-supervised learning makes use of unlabeled images of elephants and rhinos to improve the performance over supervised learning.
- Transfer learning makes use of labeled images of horses and goats. Since, the generative distribution of this dataset is expected to be similar to that of the target dataset, the representation learnt in the first task can be transferred to the second task.
- Self-taught learning utilizes unlabeled images which may have a different generative distribution than the target dataset.

3.2 Convolutional Deep Belief Networks

Deep Belief Networks are a class of deep neural-networks, composed of numerous hidden layers with connections across layers and no connections between neurons in the same layer. The connections between two hidden layers are represented using convolution matrices. Figure 2 shows connection between a single unit in hidden layer with multiple units in input layer of a CRBM. Such CRBMs are stacked on top of each other to obtain a Convolutional deep belief network. First, the CDBN is trained using unlabeled images like in Pre-Training phase for DBNs. Then, once the weights (that is, the convolutional masks) are initialized then image for which features need to be extracted is passed as input. Then the activation values of different units in hidden layers represent features for the image used as input. These features can be unrolled into a vector to be used in supervised classification tasks.

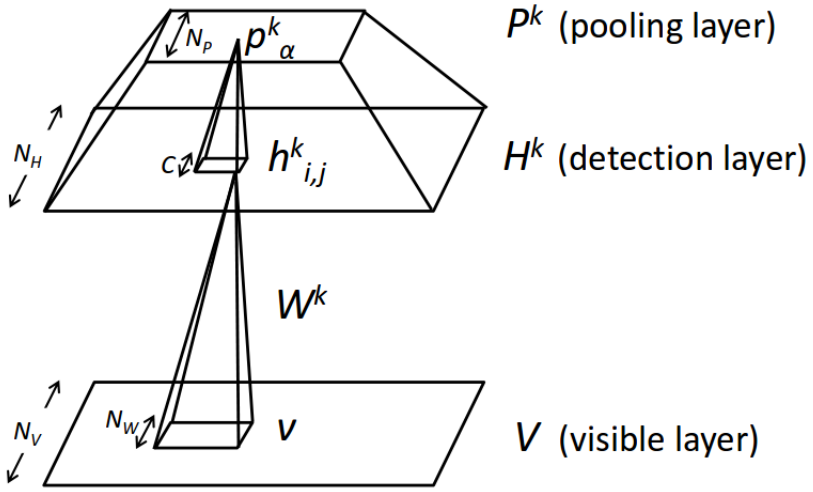

Figure 2: Convolutional restricted Boltzmann Machine with max pooling *Source: Lee et al, [1]*

4 Dataset

For the unlabeled images, we first considered the Kyoto Natural Images dataset[7] as used by Lee et al. [1] in their paper, but we had to obtain to from web archives since the authors have withdrawn the dataset due to some reasons. So, we used the ImageNet[4] dataset as it was also bigger than the Kyoto dataset allowing us to train our model on more unlabeled images. 200 images were selected randomly from the natural objects class of ImageNet. The natural images were cropped to standard size of 150*200 (Images were of natural scenes and very generic, so cropping was not altering the nature of images in anyway. Rescaling would change aspect ratios of natural features.) These were used for unsupervised training of the CDBN. For classified images, we used the Caltech101 [8] dataset. The objects were chosen from the following categories -

1. Car
2. Elephant

3. Leopard
4. Motorbike
5. Airplane

We used 20 images from each class to train our supervised model. These images were rescaled to 150*200 and then used. For testing purposes, we took 40 labelled images **(changed after poster presentation as recommended by MS Ram)** of each class to maintain uniformity (some classes had only 64 images) as otherwise accuracy would have been biased because of more data from specific classes.

5 Methodology

Since we wanted to compare performance of Self Taught Learning with normal Supervised Learning, we have considered two models. In the first case, we do not make use of the abundant unlabeled data available to us and simply train the CDBN using the labeled images as unlabeled. Then, features for these images are obtained from the CDBN and augmented to respective image vectors. Then, a multi-class SVM from the libSVM library[7] is trained using the augmented image vectors and their labels using linear kernel. **We were earlier using features of both layers but later used only the second layer which significantly reduced the input dimension to the SVM and improved the performance (changed after poster presentation)**.
In the second case, we use self taught learning to make use of the unlabeled data available to us. So, the model was first trained on the natural images from ImageNet dataset. Then, once the weights are initialized in the model using the unlabeled data, the model is used to extract features for images of the labeled dataset. For each image, the activation values of different units in hidden layers is taken and augmented to image vector to be used in supervised classification. After this process, the augmented images are used to train the SVM.

5.1 Architecture of CDBN

The Input layer dimensions for CDBN were $150 * 200$. The convolutional weights matrix used everywhere were of $11 * 11$. First hidden layer had 40 units and the next hidden layer had 100 units each having a structure as shown in 2. The pooling ratio was 2 in both hidden layers. The dimensions of a single unit in final layer was $30 * 42$.

6 Results

As shown in Figure 3 the second layer of the CDBN trained on natural images learnt some generic features including edges and different orientations, curves and contours.

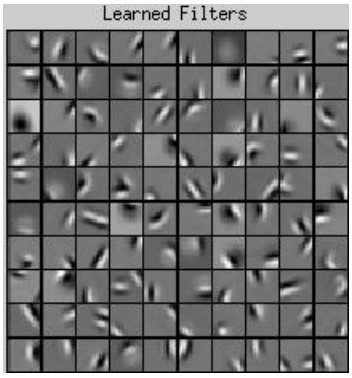

Figure 3: Features learnt by the second layer of CDBN

Self-taught learning outperformed supervised learning for our dataset. The difference is expected to become smaller as the size of training data increases.

Table 1: Results

CDBN with Self Taught Learning	**CDBN**
71%	64%

Several misclassifications occurred in the Airplane and Car categories because of the similarity between these two classes. This could also be because of presence of wheels in several airplane images.

Table 2: Confusion Matrix

	Airplane	Car	Elephant	Leopard	Motorbike
Airplane	34	0	3	0	3
Car	0	18	9	5	2
Elephant	0	21	26	0	6
Leopard	0	1	0	35	0
Motorbike	6	0	2	0	29

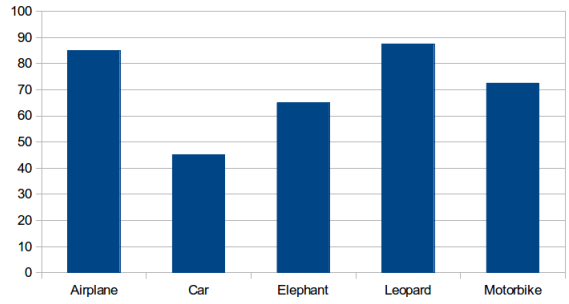

Figure 4: Classification accuracy on different classes using self-taught learning

7 Conclusion and Future Work

The domain of convolutional deep belief networks is very promising and currently, many of the benchmark results have been obtained using similar models. Self Taught Learning provides a very good alternative in situations with limited labeled data and makes use of very generic representations. These models are very complex for beginners to code and need a lot of experience for tuning them. Since, most of work on them is recent there are not many libraries available which can be used directly.[1]
The input vector dimension was a bit on the higher side for the SVM as we simply augmented the learned features. We can look into the possibility of using some non-linear dimensionality reduction techniques to reduce dimension of the input image feature vector.

8 Acknowledgements

We would like to express our sincere gratitude to Dr. Amitabh Mukherjee, our instructor for the course and MS Ram and Sunakshi Gupta, the tutors for their continuous support and guidance, which has been invaluable during the course of this project. The feedback we received in the poster presentations was very helpful and we tried to make the necessary changes in our methodology to improve our work. We would also like to thank Dustin Stansbury for the CRBM code[5].

[1]We tried several open source CDBN codes. The codes given by both H.Lee and Peng Qi were incomplete when we accessed them (around March 15, 2014). The code by A. Krizhevsky required a CUDA cluster.

References

[1] *Honglak Lee, Roger Grosse, Rajesh Ranganath, Andrew Y. Ng.* Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations

[2] *Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, Andrew Y.Ng.* Self-taught Learning: Transfer Learning from Unlabeled Data

[3] *Krizhevsky, A., Sutskever, I. and Hinton, G. E.* "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada

[4] Imagenet http://image-net.org

[5] Dustin Stansbury, Convolutional Restricted Boltzmann Machines code https://github.com/dustinstansbury/medal

[6] CalTech 101 Dataset http://www.vision.caltech.edu/Image$_Datasets/Caltech101Kyotoimagedatasethttp$: /

[7] Chang, Chih-Chung and Lin, Chih-Jen, LibSVM http://www.csie.ntu.edu.tw/ cjlin/libsvm

[8] L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE. CVPR 2004, Workshop on Generative-Model Based Vision. 2004