Scalable Deep Learning for Image Classification with K-Means and Logistic Regression

venkata K.R sanepalli
vsanepal@stanford.edu

With lot of research and advancement of deep learning, complex unsupervised learning is applied for extracting deep hierarchies of features especially to images. But, off-the-shelf unsupervised learning algorithms combined with deep learning techniques would yield results similar to complext,time consuming Deep learning algorithms.

In this report, I would use K-means algorithm based on [1][3] and apply logistic regression to NORB and CIFAR datasets using single-layer network to classify images.

1. Introduction

This project is implemented in R. I will be using spherical kmeans from [4] for feature extraction. I will be extracting sample patches from images and run shperical k-means on those patches. Since non-linear representation requires deep nueral networks or hard to understand SVM non-linear kernal, we will represent features suitable for linear classification.

Next section describes data pipeline flow and details of each of the component. Section 3 describes classification model results.

2. Data pipeline

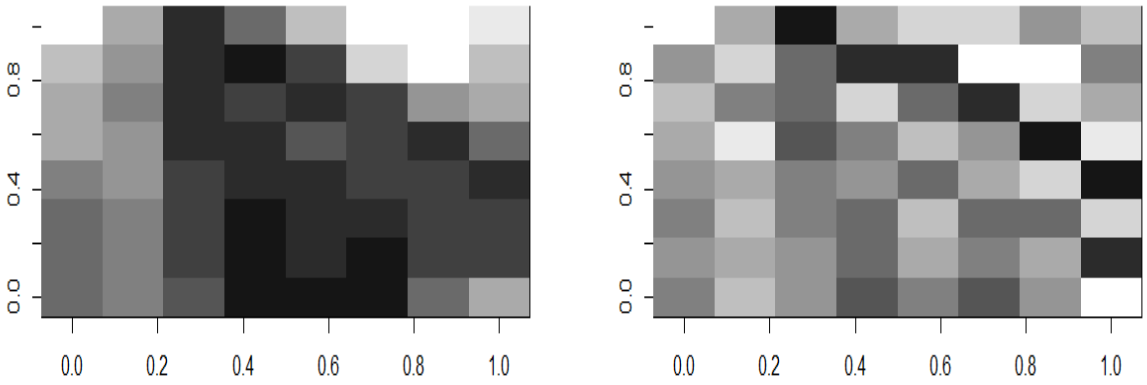
Here, I describes steps for data processing and classification.

1.Pre processing

After loading images into R, images needs to be converted to RGB pixels. I have loaded 50000 CIFAR 32X32 images and converted each of the pixels into RGB data frame.

It is common practice to perform several simple normalization steps before attempting to generate features from data. In this work, we assume that

every patch x(i) is normalized by subtracting the mean and dividing by the standard deviation of its elements. For visual data, this corresponds to local brightness and contrast normalization. After normalizing each input vector, the entire dataset X may optionally be whitened [5]. While this process is commonly used in deep learning work (e.g., [6]) it is less frequently employed in computer vision. Below left is image is before whitening and right image after whitening.



2. Extract sample patches

To create K-means cluster , I have taken 500000 8X8 size patches from 50000 CIFAR images . Each of those are an image of CIFAR training data set. All the patches are selected in random location of a image. Image is selected randomly.

3. Extract Features

In this step, I extract image features. Instead of looking for entire data set for features. I will be following suggestions in [1] and [2]. I tried to use off-the-shelf spherical K-means cluster for patches extracted in step2. But, this took forever to finish. So, I used spherical k-means alogorithm code from open source. the standard spherical k-means problem is to minimize the below

$\Sigma_i(1 - cos(x_i, p_{c(i)}))$

Spherical K-means steps as follows.

1. Normalize inputs:

$x^i := \frac{x^i - mean(x^i)}{\sqrt{var(x^i) + \epsilon_{norm}}}, \forall_i$

2. Whiten inputs:

$[V, D] := eig(cov(x)); //$ So $VDV^T >= cov(x)$

$x^{(i)} := V(D + \epsilon_{zca} I)^{-1/2} V^T x^{(i)}, \forall_i$

3. Loop until convergence (typically 10 iterations is enough):

$s_j^i := \{ \ D^{(j)T} x^{(i)}$
0

For this setup I run it with 1,600 centroids

4. Mapping the original features (pixels) of the dataset into a new feature space, which is a function of the learned dictionary

separable. As suggested in [1], I implemented the soft threshold encoding scheme for K-Means. In this step, each original 32 x 32 pixel image, represented in a 1 x 1024 vector, is now encoded in a 1 x 6400 vector.

5. Classify the data set, now represented in the new learned feature space, using standard linear classification algorithms

In this excercise, I have used soft max regression for classification with batchs of 50 sample samples.

Results

The CIFAR database of images, used to test this framework, is composed by 60,000 labeled training images and 10,000 labeled test images. K-Means was tested using 500,000 8x8 patches extractedfro m all available training data set. The encoding and classification steps used 30After running an Munltinomial classifier, I got the results depicted by the following confusion matrix: row are predicted and columns are true labels

This result corresponds to a test error rate of 2.7 , which is better than all results published for 2 or 3-layer neural networks trained without cross-entropy loss function and without deskewing pre-processing. It suggests that, if training the system with 100 of the data set, and applying deskewing to it, it could improve significantly.

columns: TRUE labels

rows: prediction

	1	2	3	4	5	6	7	8	9	10
1	272	0	0	1	0	0	2	1	0	2
2	0	335	1	0	0	0	0	2	1	0
3	0	1	283	4	0	1	0	0	3	0
4	0	0	1	327	0	1	1	3	1	0
5	0	1	1	0	295	0	5	0	3	2
6	3	0	2	7	0	263	3	0	2	0
7	1	0	0	0	0	3	298	0	0	0
8	0	1	1	0	0	0	0	315	2	1
9	2	0	0	1	0	4	0	0	262	2
10	0	1	0	0	3	0	0	4	0	269

References

[1] Coates, A., Ng, A.Y. Learning Feature Representations with K-means, G. Montavon, G. B. Orr, K.R. Muller (Eds.), Neural Networks: Tricks of the Trade, 2nd edn, Springer LNCS 7700, 2012

[2] Coates, A., Lee, H., Ng, A.Y.: An analysis of single-layer networks in unsupervised feature learning. In: 14th International Conference on AI and Statistics. pp. 215 223 (2011)

[3] vilcek,alexandre Scalable Deep Learning for Image Classification with K-Means and SVM

[4] https://cran.r-project.org/web/packages/softmaxreg/softmaxreg.pdf

[5] A. Hyvarinen and E. Oja. Independent component analysis: algorithms and applications. Neural networks, 13(4-5):411430, 2000.

[6] M. Ranzato, A. Krizhevsky, and G. E. Hinton. Factored 3-way Restricted Boltzmann Machines for Modeling Natural Images. In AISTATS 13, 2010