Scalable Deep Learning for image classification with K-means and Logistic Regression

V.Krishna.R Sanepalli

Stanford University

Abstract

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 complex,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.

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.

General Approach

1. pre-process data.
2. Apply spherical k-means for feature extraction of images.
3. Map original features into new feature space learnt from k-means. This step should have turned features into linearly seperable.
4. Apply multinomial logisic regression for classifying images .

Materials and Methods

the standard spherical k-means problem is to minimize $\Sigma_i(1cos(x_i, p_{c(i)}))$

over all assignments c of objects i to cluster ids c(i) 1, . . . , k and over all prototypes p1, . . . , pk in the same feature space as the feature vectors xi representing the objects.

The softmax function is used in various multiclass classification methods, such as multinomial logistic regression (also known as softmax regression), multiclass linear discriminant analysis, naive Bayes classifiers, and artificial neural networks. Specifically, in multinomial logistic regression and linear discriminant analysis, the input to the function is the result of K distinct linear functions, and the predicted probability for the j'th class given a sample vector x and a weighting vector w is:

$$P(y = j|X) = \frac{eX^T w_j}{\Sigma_{k=1}^{K} eX^T w_k}$$

Mathematical Section

Initialization, pre-processing and full K-means training could be represented as follows.

1. Normalize inputs:

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

2. Whiten inputs:

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

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

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

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

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 7x7 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.

TRUE labels

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

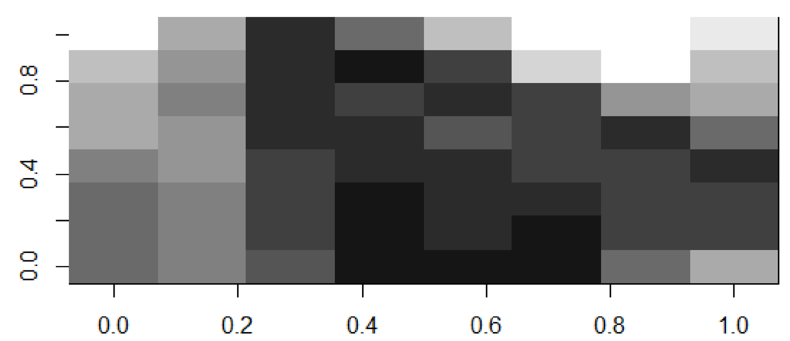

Figure 1: Before patch pre-process

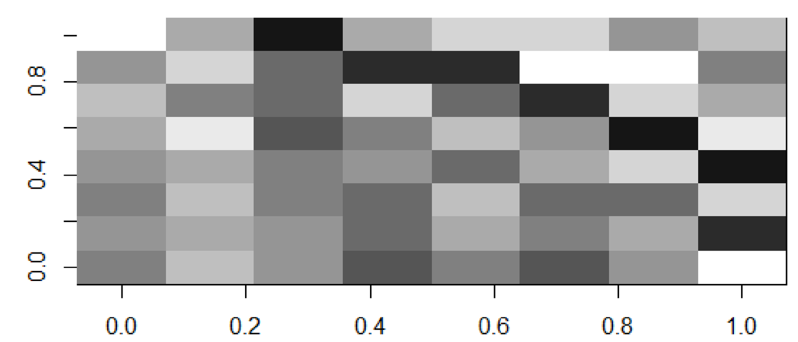

Figure 2: Afte patch pre-process

Conclusions

- Using k-means algorithm to extract feature and applying multiclass logistic regression has similar performance to using SVM. training on entire data dataset in a single machine is time consuming.
- Using off-the-shelf spherical k-means did not help as it took forever to complete.

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/skmeans/skmeans.pdf