Unsupervised features learning for image classification

Fella Belkham Ahmed Mazari Hafed Rhouma

December 2, 2016

Abstract

Recently, image classification draw attentions of many researchers. The need of object recognition grows drastically, especially in the context of biometric, biomedical imaging and real time scene understanding. Computer vision task is the most challenging in machine learning. For that reason, it's fundamental to tackle this concern using appropriate clustering and classification techniques. However, the quest for the best unsupervised features extraction remain an open problem even if CNNs reach a remarkable success, establishing new state-of-the-art. In this context, we study from an acute insight standpoint the standard clustering models K-means, GMM and Naive Bayes classification algorithm in order to draw conclusion and underline their limits for such complicated tasks.
To what extent are k-means and GMM efficient ? Why they fail and how to circumvent their weaknesses.

Keywords : Unsupervised learning, Dimensionality reduction, Classification, Gaussian Mixture Models, K-means, Naive Bayes

1 introduction

Videos, photos ,cameras are appearing everywhere : cameras on satellite, cameras taking medical images. The target to reach is to be able for instance to build accurate medical cameras for diagnosing, detecting tiny malignant cells in the brain to tackle brain cancer.
Computer vision responds to the most challenging problem in our daily life in different domain. Solving this kind of problem allows to better up the human being life and reduce cancers.
Computer vision tries to mimic human vision but the goal is very long way. So understanding the most complex and rich human sense appeals machine learning student and researchers. Over half the human brain is involved in processing visual information (we are processing 60 images per second with millions of pixels in each image).
From a machine learning standpoint, computer vision is the most challenging problem because it aggregates several complex tasks such as scene understanding which interfere with natural language processing. The thing that makes computer vision task difficult is the fact that we have thousands, millions of parameters and we need a huge dataset to solve efficiently the problem.
In this project we implement and evaluate the performance of K-means [3] and GMM clustering on CIFAR-10 dataset [1] for features extraction. We first, extract patches from each image. Then, we apply normalization and ZCA [2] on the patches. After that, we train K-means hard and GMM to extract features. Furthermore, we use the extracted features to classify cifar-10 [1] images with Naive bayes. Finally, we evaluate and discuss the results.

2 CIFAR10 Dataset

The CIFAR-10 is a smaller subset of the large 80 million tiny colour images dataset of the size (32x32). It has the advantage of having everything labeled in 10 different categories airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck.
CIFAR-10 contains 60,000 images divided into six batches. Five of them are useft for training, in which, the images are in random order, some training batches may contain more images from one class than another. The last batch is used for test, it contains exactly 10000 randomly-selected images from each class.

3 Preprocessing

For the preprocessing phase, we apply normalization and whitening to our data. Each patch is normalized by substrcting the mean and dividing by the standard deviation of its elements. Whitening is optional, but it's important to apply it [2] before applying kmeans clustering [3] because it scales all the variances such that the covariance matrix for a transformed set of patches is the identity matrix by removing second order correlation from each dimension of data. It also enables the use of a simple feature learning algorithm over more complex technique such as GMM and Kmeans. Furthermore it empirically boosts final classification accuracy regardless of feature learning technique [2].

4 Feature extraction

A set of techniques in machine learning that learn a transformation of raw inputs to a representation that can be effectively exploited in a supervised learning task such as classification. The learning process can be either supervised or unsupervised and can be done by using multiple algorithms like: – Autoencoders, Clustering, Neural Networks [4].
The main use of feature learning in our project is to use it for classification purpose. In the feature learning process, we start by extracting random patches from our data, note that the number of patches to extract is unknown, so it

can be used as a parameter and it will be interesting to show how it affects the classification process. After applying the per-processing on each patch, we learn feature using an unsupervised algorithm, at this stage, we tried clustering with K-means Hard and GMMs. The next step consist of using the learned feature for classification, to do so, we start by reducing the dimensionality of each image in the dataset:

- Extract w-by-w patches separated by s pixels (strides)
- Using one of the chosen unsupervised algorithm, we map each patch p ∈ Rn to a new vector p' ∈ Rk using the function f that corresponds to the algorithm.
  For example, in the k-means hard, each centroid can be used to represent a feature. In tha case, our new vector p' will be a binary vector, with only one value at 1 at the ith position, if it is closest to the ith centroid learned by k-means.
- Reconstruct a new representation of every image by reducing its dimensionality to (n-w+1)-by-(n-w+1)-by-K using the resulting p' vectors, where K is a parameter of the algorithm (the number oc clusters).
- For each image, pool the vector into 4 equal size quadrant and compute the sum over them to obtain the classifier features of the length 4K. The features obtained through this process are representations of the images that will be used for classification.

5 Expirement and results

We started the experiment by applying Naive Bayes on the raw data, in order to compare it to the results obtained by it on MNIST dataset. This will allows us to see the impact of RGB images compare to grayscale images on the performance of the classifier. Furthermore, the complexity of CIFAR-10 images can be axplained by the fact that it contains overlapping objects.

Then, the second step of the expirement consists of applying the feature extraction process for classification purpose.
We applied both K-Means Hard and GMM with different configurations. We trained Kmeans hard on the extracted patches using scikitlearn package with a non random initialization and k varies from 50 up to 300 such that k represents the number of features. After predicting the classes of each new patches, we create a new representation of each image. This new image representation [3] is used to classify cifar-10 [1] images with Naive bayes. To evaluate the performance of K-Means, we adopted Silhouete Score, Elbow Curve (Avrage within cluster sum of squares) and Percentage of variance explained criteria. We trained a Gaussian Mixture Model using mixture package from scikitlearn, with ncomponent=100 and covariance-type="full" as parameters. To evaluate the performance of GMM, we adopted AIC and BIC criteria. After predicting cluster belongings of new patches to classify, we got the new images representations and applied our naive Bayes classifier.
To sum up, our parameters where the number of clusters k, the number of randomly extracted patches, using whitening We obtained the following results: We define N = number of patches initially extracted from the image to train the GMM model.

	K	Random Patches	Whitening	Normalization	Accuracy
Naive bayes	/	/	/	/	**27.6**
GMM	100	16	Oui	Oui	18,2
Kmeans hard	50	32	Oui	Oui	**15,83**
	50	16	Non	Non	**26,13**
	100	32	Oui	Oui	11,33
	100	24	Non	Non	**26,93**
	300	16	Oui	Oui	10.13

Figure 1: Classification accuracy using Naive Bayes

As presented in Figure 1, the results are surprisingly negative, especially the fact that the results without normalization and whitening are better then when we apply them. The feature extraction process did not perform as predicted according to [3]. The main difference that could explain our results is the choice of the classifier, Naive Bayes only reach 77% on the MNIST dataset, while the benchmark results on MNIST reaches 99.79% accuracy using DropConnect Regularization Neural Network [6].

6 Conclusion and perspectives

This study allowed us to get acquainted with images clustering and classification. We noticed that the choice of the appropriate model depends on the complexity of the problem and there is no algorithm better than the other without empirical studies.
In perspective, we envisage to study fractional max pooling CNN which is the-state-of-art performance on Cifar-10 classification (96.53%) [5]. Furthermore, we cannot extrapolate our results since we only explored only one batch to extract the features

References

[1] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton.The CIFAR-10 dataset.

[2] Derek Rose, Itamar Arel .Toward a Sequential Approach to Pipelined Image Recognition.

[3] A.Coates, H.Lee, A.Y. Ng.An Analysis of Single-Layer Networks in Unsupervised Feature Learning

[4] R. Johnson. Feature Learning slides presentation.

[5] Benjamin Graham.Fractional Max-Pooling

[6] Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, Rob Fergus. Regularization of Neural Networks using DropConnect