Final Report for 10601

Jinhong Chen jinhongc@andrew.cmu.edu

Da Wang dawang@andrew.cmu.edu

Abstract

This is the midterm report for project of Introduction to Machine Learning(10-601, Fall 2015). In this project, we perform several machine learning method to classify images on the CIFAR-10[1] database. Conducted by an empirical study, the goal is to investigate the performance of different classifiers with different features and Principle Component Analysis(PCA). We also test deep learning algorithm with unsupervised feature extraction. Currently, by using neural network with hog feature, we can obtain a testing accuracy of 52%, which is 4% higher than the baseline implementation.

1 INTRODUCTION

Classifying images to different categories is the main focus for automatic recognition. In order to achieve higher precision, various methods of computer vision are proposed. Nowadays, deep learning is one of the most popular and effective machine learning techniques in this domain because of its ability of modeling image data with high-level abstractions[2]. The number of dimensions of deep learning algorithm can be extremely large which is barely possible for human to understand. However, traditional computer vision features such as GIST, Scale-invariant Feature Transform(SIFT), Speeded Up Robust Features(SURF), HOG are also significant for accurately image classification. Instead of using the raw pixel data, the features mentioned above can capture different structures such as edges, lines, orientations, etc[3].

As the main focus of these project is to test different machine learning algorithms, the precision of classification will not be the only criteria of the project. Instead, we will use different features to evaluate the performance of Logistic Regression, Naive Bayes, Neural Network and Support Vector Machine. Besides, we also use Principle Component Analysis(PCA) to reduce the dimensions of features to improve accuracy. In order to find the best parameter for each machine learning algorithm, we tuned the parameters with many options and then compare their performance.

Another work for this project is building a data preprocessing pipeline so as to eliminate outliers in the training data. The idea of this method is to use better training data to get better classifier. So we cover all three factors that may affect the precision of classification. Data preprocessing contribute to better training data. Different feature extraction and PCA method contribute to more effective descriptor. Testing different machine learning algorithm can find the best machine learning technique that meets our needs.

2 BACKGROUND

This section introduce the dataset, feature and machine learning algorithm related to this project.

2.1 Dataset

The CIFAR-10 is a labeled subset of 80 million tiny images which consists of 60000 32x32 color images in 10 classes, with 6000 per class. There are 50000 training images and 10000 test images. Below is a random sample from the CIFAR-10 with 10 random images from each class.

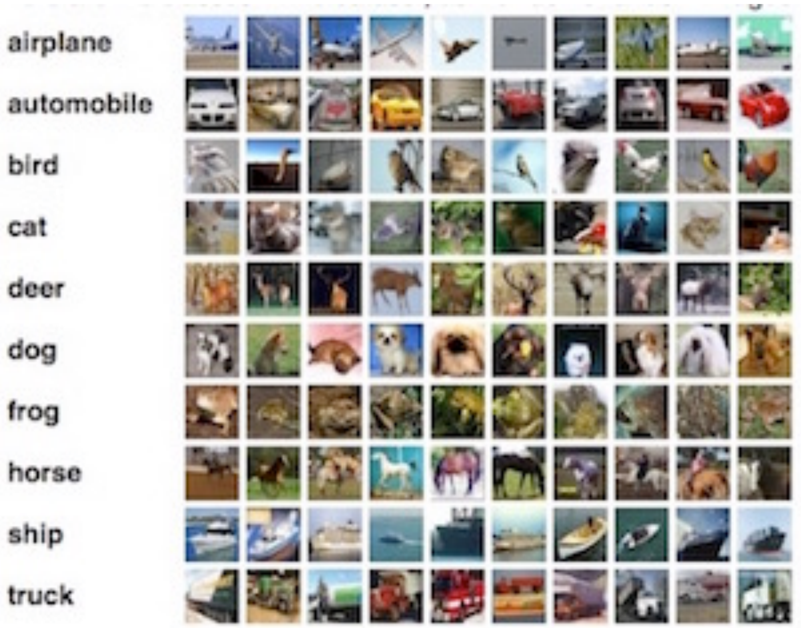


Figure 1. a sample of images in CIFAR-10

2.2 HOG feature

In computer vision, it is critical to find proper visual features so that complex tasks can be performed. In this project, it is better to use the features of the image as they are more descriptive than the raw image pixel. Compared with SIFT which captures the properties at key points, HOG describes the shapes of a given region in a broader scope. Also, HOG is typically used in a sliding window fashion in object detection systems. The images in CIFAR-10 are quite small(32x32) so it may be better to use HOG in this project.

2.3 SIFT feature

SIFT computes the gradient histogram only for patches (usually 16*16 divided into 16 cells) around specific interest points obtained by taking the DoG's (as an approximation to LoG's) in the scale space. It is a local descriptor which makes it perform bad when the input images are small.

2.4 GIST feature

GIST is typically computed over the entire image (i.e. it is a global image descriptor) for the purposes of scene classification. The idea is similar to HOG which focus on the global feature rather than the key points. So it is also a good choice to use GIST features.

2.5 Logistic Regression

Logistic regression is one of the most useful linear classifier in machine learning. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution. The logistic function is shown below:

$$a = \frac{1}{1 + exp(-b)}$$

We can use Maximum Conditional Likelihood Estimation(MCLE) to train the model.

2.6 Neural Network

Unlike logistic regression, neural network with multiple hidden layers can be regarded as non-linear classifier. It has been proven to be powerful in many domains especially image classification. Typically, a neural network can be composed of a large number of interconnected computing elements in order to learn high level features from the input layer. An example of the structure of a neural network is shown below:

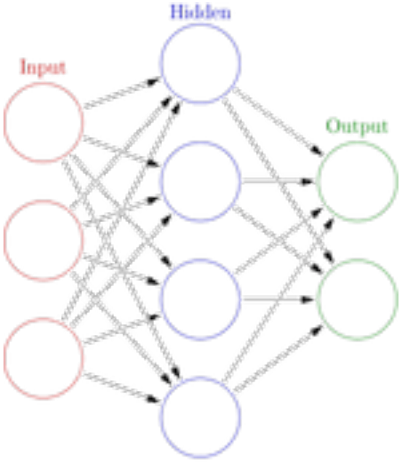


Figure 2. a sample of neural network with one hidden layer

However, neural network cannot provide a explicit learning process of its hidden layers. In order to find the best parameters, we usually increase the number of hidden layers which takes more time to train the neural network. After comparing the performance under different parameters, we can select the best neural network model.

2.7 Support Vector Machine

SVM is a discriminative training process of linear classifier by maximizing the margin hyperplane of classification[4]:

$$f(x) = \sum_{i=1}^{L} \alpha_i t_i K(\mathbf{x}, \mathbf{x}_i) + d_i$$

As CIFAR-10 dataset contains 10 different classes, we need to use the multi-class SVM to classify the images. Using kernel functions, SVM can often find the proper linear spreadable hyperplane in higher dimension while the data is not linear separable originally. We have tested several kernel function and it turns out that the Radial Basis Function(RBF) kernel performs better in this project.

$$K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \gamma > 0$$

Here γ is the kernel parameter. The optimal value of γ should be selected by cross validation method.

2.8 Principle Component Analysis(PCA)

PCA refers to a specific form of dimension reduction where the principle components are drawn on the sequentially orthogonal axes of the largest variance. It provides a way to reduce dimensionality without losing too much information[5]. With the help of PCA, we can build a model without redundant or irrelevant features.

3 METHOD

3.1 Feature Extraction

In the project so far, we tried two kind of feature extraction method before classification. One is the raw intensity of the image, the other is the Histogram of Oriented Gradient (HOG).

3.1.1 Raw Intensity

When using the raw intensity as the feature, we did not extract feature from the images. Each image (row) in the testing data X was treated as the feature of the image, and input directly to the classifiers. The raw intensity, as a kind of feature, represent the isolated color value of the pixel in the images.

3.1.2 Histograms of Oriented Gradient (HOG)

The HOG feature was extracted from the images using a open source library (VLFeat) by first transforming each image (row) to a matrix and calling the 'vl_hog' routine.

The HOG feature extract the orientation feature from the image by counting the occurrences of gradient orientation in local patches of the images, and place them in discrete bin which forms the histogram of gradients. The magnitude of the gradient is used as the vote on the histogram. The HOG feature captures the information of edges (oriented gradients) in the images, since gradient near the edge will have large magnitude.

3.2 Classifiers

Second level headings are lower case (except for first word and proper nouns), flush left, bold and in point size 10. One line space before the second level heading and ½ line space after the second level heading.

3.2.1 Naïve Bayes Classifier

To implement the Naïve Bayes Classifier, we optimized the classifier we implemented in Homework 1 by vectorizing calculation both in classification.

In training stage, the classifier calculates mean and variance of each feature in each class. Each feature in each class is considered a Gaussian distribution independently. And later in classification stage, the classifiers calculate the probability of the input sample belonging to each class, and pick the class which maximize the probability to be the result.

3.2.2 Logistic Regression (Softmax)

Since the Logistic Regression technique can only classify binary target. We extended our Logistic Regression to Softmax Regression, by using a 3rd party function optimizer (minFunc) to find the parameters which minimize the error. The Softmax regression is a generalized version of logistic regression in that it estimates the probability of a data belonging to a class in a way similar to logistic regression (using sigmoid) but in the meantime it supports multi class classification by adding more sigmoid units. The hypothesis made by the Softmax regression is shown in the following equation:

$$h_\theta(x) = \begin{bmatrix} P(y = 1|x;\theta) \\ \dots \\ P(y = 10|x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta^{(j)T}x)} \begin{bmatrix} \exp(\theta^{(1)T}x) \\ \dots \\ \exp(\theta^{(10)T}x) \end{bmatrix}$$

Where P(y=i|x;θ) is the probability that the input x classified to class i, and θ is the parameter to the regression. To find the parameter, we have to minimize the cost function J(θ) over training data. We minimize the cost function using a third party optimizer `minFunc` in our implementation.

$$J(\theta) = -[\sum_{i=1}^{m} \sum_{k=1}^{K} \delta(y^{(i)} = k) \log(\frac{\exp(\theta^{(k)T} x)}{\sum_{j=1}^{K} \exp(\theta^{(j)T} x)})]$$

3.2.3 Neural Network

We implemented our neural network using Feedforward and Back propagation algorithm with an adjustable size of hidden units with 3 layer (input + hidden + output). In the neural network classification, we used the HOG feature as the input to the neural network.

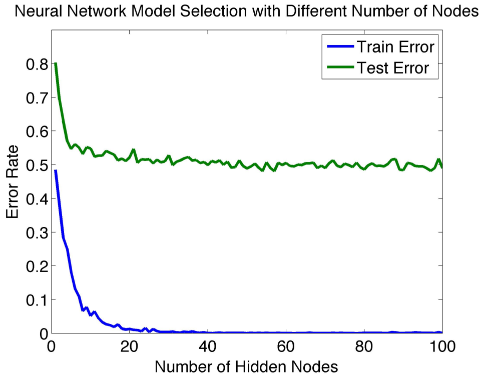
3.2.3.1 Features selection

Since the features of HOG has a dimension of 1984, which is pretty large resulting in a large scale of parameters in the neural network ,we decided to first filter out some features using the following policy: beginning with the first feature, we incrementally add a new feature into the feature set, if the resulting classification (using Naive Bayes Classifier) accuracy increased, we keep this feature, otherwise, we abandon this feature. After applying this policy, we selected 760 features from the HOG feature.

3.2.3.2 Model selection

The number of nodes in the neural network has great influence on the number of parameters in the model. Too much parameter can result in overfitting. In our case, the input feature is 760 dimension, and the number of output class is 10. Adding a node in the hidden layer adds 761+10 parameter to the model.

To prevent the resultant overfitting by setting too much nodes in the hidden layer, we trained and test(test on the 'test_batch.mat') the neural network by incrementally adding nodes to the neural network, this give use the error rate on both the training set and test set with different number of nodes in the hidden layer which is shown in the following plot.



Finally, we chose 25 nodes for our hidden layer, because the test error rate cease to reduce significantly with new nodes added. This gives us an accuracy of 48.4%, which is slightly different from what we got during the midway report, the reason is the initial weight is initialized randomly each time a neural network is created, so they ends up a different resulting parameters after training.

3.2.4 Extreme Learning Machine

We implemented our extreme learning machine based on [6]. In the extreme learning machine classification, we used the HOG feature as the input to the extreme learning machine. Also, we have tried different kernel function to test the classifier: sigmoid, sin, hardlim, tribas and radbas.

3.2.4.1 Features selection

Since the features of HOG with cell size 4 has a dimension of 1984, which is pretty large resulting in a large scale of parameters ,we decided to use a larger cell size (which is 8) so that we get 496 dimension features for a single image. Accompanied with the decrease of the number of dimensions for the feature, the interference among features decreases in the same time. We've tried different cell size such as 12, 16 but the results show that 8 is the proper size for this problem.

3.2.4.2 Model selection

The number of nodes in the extreme learning machine has great influence on the number of parameters in the model. Too much parameter can result in overfitting. In our case, the input feature is 496 dimension, we've test different number of neuron nodes as the hidden layer and the result shows that 1400 hidden nodes is the best choice

3.2.5 Linear Discriminant Analysis

We implement our Linear Discriminant Analysis algorithm based on [7]. We us HOG feature as the input of the LDA classifier. LDA explicitly attempts to model the difference between the classes of data, so it can also be a multi-class classifier.

3.2.5.1 Features selection

Since the features of HOG with cell size 4 has a dimension of 1984, which is pretty large resulting in a large scale of parameters ,we decided to use a larger cell size (which is 8) so that we get 496 dimension features for a single image. Accompanied with the decrease of the number of dimensions for the feature, the interference among features decreases in the same time. It greatly increases the accuracy for prediction because of each feature can be more descriptive.

4 RESULT

Feature+Classifier	Accuracy
Raw+NB	29.1%
Raw+LR	26.2%
HOG+NB	47.6%
HOG+LR	45.5%
HOG+NN(3L+25N)	48.2%
HOG+ELM(6400)	51.4%
HOG+LDA	51.4%
Others(sfm, svm)	< 50.0% (included in the hand-in package)

Table 1. Comparison on different feature/classifier combination

5 CONCLUSION

We've tried different machine learning algorithms with different features to achieve higher accuracy. The great difference between Raw image data and HOG feature show that these features are high representative and useful. On one hand they reduce the computation complexity, on the other side they make us easy to compare among different algorithms.

The basic machine algorithms, such as naive bayes and logistic regression, are easy to implement but with limited ability to classify images. Other methods such as neural network and extreme learning machine, can have totally different performance with different parameters. As it is, we have to spend lots of time tuning all the possible combination of parameters to find the best solution. Mathematical method, such as linear discriminant analysis, is highly efficient if the proper features are chosen.

For other algorithms we have tested, for example support vector machine, are originally binary classifier thus have different strategies to become the multi-class classifier.

However, there are still several algorithms that we don't have enough time to implement such as decision tree, adaboost, etc. The most important thing we learned from this course is that though it is important to know the detail of the machine learning algorithm, for real life usage, there are lots of other factors will affect the performance of different algorithms. We need to practice more to build our instinct and sensitivity for solving different problem with different techniques.

Acknowledgments

Thanks for the patience of our teacher and TAs. They try their best to help us have a deeper understanding on what we learn and how to use them in real life project.

References

[1] Alex, K. 2009. Learning multiple layers of features from tiny images

[2] Honglak, L., Roger, G., Rajesh, R., and Andrew Y. Ng. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. ICML'09,609-616

[3] Andreal, V., and Brian, F. 2010. Vlfeat: an open and portable library of computer vision algorithms. ACM, New York, NY 2010, 1469-1472

[4] Leslie, C., Eleazar, E., and Stafford, W., 2001. The spectrum kernel: A string kernel for SVM protein classification. Pacific Symposium on Biocomputing. 566-575

[5] Morre, B., 2009. Principal component analysis in linear systems: Controllability, observability, and model reduction. Automatic Control, IEEE Transactions on Vol. 26. Issue. 1.

[6] Guang-Bin Huang, 2015. What are Extreme Learning Machines? Filling the Gap Between Frank Rosenblatt's Dream and John von Neumann's Puzzle. Cogn Comput (2015) 7:263-278

[7] Martinez, A. M.; Kak, A. C. (2001). "PCA versus LDA" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (=2): 228–233. doi:10.1109/34.908974