Comparing and Collaborating Classifiers for Multiclass Image Classification

San-Chuan Hung
sanchuah@andrew.cmu.edu

Jiajun Wang
jiajunwa@andrew.cmu.edu

ABSTRACT

Many image object recognition methods need machine learning models as the basic training algorithm. In this project we compared the performance four well known machine learning models: Neural Network, Random Forest, Multinomial Logistic Regression, and Support Vector Machine. Furthermore, we analyzed the accuracy of different labels in each classifier, and found that each classifier has its strong points and weak points. We showed that ensemble methods, aggregating different model prediction results, can improve the performance from 0.76% to 13.07%.

1. INTRODUCTION

Multiclass Image Classification is to classify images based on image data to multiple classes. Given the pixel values in RGB channels of an image, the machine has to discriminate the image as a bird, a cat, or a truck. To solve this problem, a common approach is developing multiple computer vision decriptors to help machines to learn how to discriminate different objects, like SIFT [4], and feed the image features into a machine learning algorithm to learn a classifier. In this approach, because no matter what kind of image features are developed, it still needs a machine learning classifier in the final step. Machine learning algorithms are important basement.

In this paper, we mainly compared 4 well-known classification algorithms: Neural Network [2], Random Forest[1], Multinomial Logistic Regression[3], and Support Vector Machine[6]. First, we systematically tuned each classifier parameters to its best performance. Second, we compared and analyzed their prediction results: What is the best classifier for the tasks? What are strong labels and weak labels for each classifier? Last, we came up an ensemble method to aggregate prediction results from different classifiers.

2. BACKGROUND

2.1 Multinomial Logistic Regression

Multinomial Logistic Regression (MLR)[3] is a machine learning model extending Logistic Regression from binary class classification to multi-class classification. Similar to Logistic Regression, MLR learns a vector of linear model parameters W_k of X for each label k, and uses sigmoid function to generate the probability of each label. We used the implementation of MLR in provided matlab code, and tuned the parameter of cost and the gamma value of RBF kernel.

2.2 Support Vector Machine

Support Vector Machine(SVM)[6] is a machine learning model to find a boundary to maximize the gap between different class instances. We used the implementation of SVM in provided matlab code, and tuned the parameter of cost and the gamma value of RBF kernel.

2.3 Radial Basis Function Kernel

Radial Basis Function Kernel(RBF Kernel)[5] is a kernel trick to convert origin feature samples into input space where . We used the implementation of provided matlab code, and tuned the parameter of gamma.

2.4 Multilayer Neural Network

Neural network [2] is a machine learning model that mimic human neural system. Generally a neural network maps input data onto a set of desired outputs. The network contains multiple layers of nodes called hidden layers. The nodes are connected as a directed graph. Nods in each layer are connected to the nodes in next layer. Each node in the hidden layers is with an activation function. In order to train these activation functions (the neuron), neural network usually utilizes gradient descent and backpropagation methods. The advantage of multilayer neural network is that it usually achieve a higher test accuracy than other classifiers. The disadvantage is that training a network requires lot of computing resource and time.

In this project, we used fully connected neural network model. The description of the classifier class we used: NAME: weka.classifiers.functions.NeuralNetwork SYNOPSIS(Convolutional) Neural Network implementation with dropout regularization and Rectified Linear Units. Training is done with multi-threaded mini-batch gradient descent.

2.5 Random Forest

Random forest [1] is an ensemble methods that classifies based on multiple decision trees. The trees are constructed during the training time with random seeds and the final output is based on the voting. By using the method, the classifier will not be highly sensitive to noise in the training data. So the model has a better accuracy than a single decision tree. Constructing the random forest is relatively fast. And the test accuracy is usually good. So this method is widely used.

2.6 Ensemble method

Ensemble methods combine multiple classifier to achieve better prediction result. In this project, we implemented simple voting method to aggregate different prediction model results.

3. METHODS AND RESULTS

3.1 Dataset

We used a subset of CIFAR-10[1] as our multi-class image classification dataset. In this data set, each feature vector is a 3072 dimensions data, representing a 32*32 RGB image. There are 10 types of labels to predict. Only 4000 samples with labels are given. The number of test dataset is 15000.

3.2 Preprocessing

3.2.1 Feature Expansion

At first, we expands the feature size by adding simple image features: color histogram and edge, and we compared the performance of the models trained on raw data and the models adding expansion feature. We found that feature expansion can improve performance; however, feature expansion is not this project main core, so we attached the experiment result in the appendix.

3.2.2 Color Histogram with Different Bucket Sizes

Color histogram means the distributions of values in different channels of a whole image. Also, we generated color histogram features in different bucket size. If the bucket size increases, the range of pixel value mapping to the same bucket becomes wider. We generates color histogram feature with bucket size = 32/64/128/256.

3.2.3 Edge Feature Vector

Edge feature is boundary of objects in image, generating generates a binary image whose size is as same as original image. We transformed an edge image into two feature vectors by summing up values in an edge image along x-axis and y-axis. We used matlab default function to generate edge feature vectors

3.3 Classifiers Comparison

Second, we compared different classifier paradigm: Multinomial Logistic Regression, Support Vector Machine, Neural Network, and Random Forest. We trained the classifiers based on expanded feature set, tuned classifier parameters according to cross-validation.

3.3.1 Classifier Tuning

Multilayer Neural Network

We used a neural network classifier from weka [2]. Compared with official weka class MultilayerPerceptron, this classifier runs faster and support more tuning items about the hidden layer. For tuning this classifier, we tried following parameters: inputLayerDropoutRate, weightPenalty, maxIterations, hiddenLayers, hiddenLayersDropoutRate. Note that we use default setting on learningRate. Although it is a very important parameter, we found Auto-detect leads to a better result.

[1]http://www.cs.toronto.edu/ kriz/cifar.html
[2]https://github.com/amten/NeuralNetwork

Here we list the parameters and the accuracy result with raw data input in the table 1.

From the table 1, we think increasing the hidden layer is helpful to improve prediction accuracy. However, when we tried to use more complex network, the weka complain out of memory. So 400, 40 is the best we can do. For other parameters, we tried to find out local optimize (the settings in the last column). We didnâĂŹt verify whether this setting is global optimal or not given very limited time. It turns out that multiple layers neural network can achieve the best performance among all classifiers we tried in this project. Although it needs much more time to train. Usually it takes 20-100 minutes to get a result.

Random Forest

We used weka.classifiers.trees.RandomForest from weka. The most important parameters we tried to optimize are maxDepth and numTrees. It turns out that by using more trees and allowing deeper trees, we have better predict accuracy. However, although training accuracy will increasing as we set bigger values for maxDepth and numTrees, test accuracy does not continue enhancing after maxDepth=80 and numTrees=50. For avoiding overfit, we stop tuning at this configuration. The accuracies are around 35

MLR with RBF Kernel and L2 SVM with RBF Kernel

To tune Kernel MLR and Kernel L2 SVM, we developed from the provided matlab code. First, we implemented cross-validation method in matlab, shown in âĂIJcvClassfier.mâĂİ. Second, we programmed a grid script for parameter choosing, conducting cross-validation in a grid of parameter settings and selecting the parameter set of best performance one, shown in âĂIJgrid.mâĂİ.

3.4 Single Classifier Comparison

After tuning the parameters of four classifiers, we trained them based on expanded features, and test their performance by kaggle evaluator. The result is shown Table 2. We found that Neural Network has best performance in four classifier, whose accuracy = 0.47. The performance of Kernel L2SVM and Kernel MLR have similar performance, whose accuracy = 0.45. Random Forest performed worst in this task, whose accuracy is only 0.35.

Table 2: Classifier Comparison

Classifier	Parameters	Test Accuracy
Neural Network	-wp 0.001 -mi 1200 -hl 400,40 -di 0.15 -dh 0.3	47.333%
SVM	c = 0.1, gamma = 0.01	45.848%
MLR	c = 0.1, gamma = 0.01	45.410%
Random Forest	-I 50 -depth 80	35.029%

3.5 Label Analysis

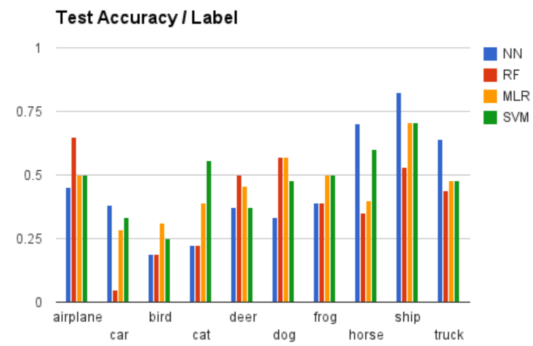
Moreover, we compared the accuracy of classifiers label by label. To do so, we manually labeled 200 images in testing data, and calculated the accuracy of each label in different classifiers. The result is shown in Figure. 1.

We found that no classifiers dominate all label classification problems. Instead, each classifier has its strong labels and

Table 1: Neural Network Parameter Tuning

inputLayerDropoutRate	0.2	0.2	0.2	0.2	0.1	0.15
hiddenLayersDropoutRate	0.4	0.4	0.4	0.4	0.2	0.3
weightPenalty	1.0E-8	0.001	0.005	5.0E-4	0.001	0.001
hiddenLayers	250,10	400,40	400,40	400,40	400,40	400,40
maxIterations	1000	1000	1000	1000	1000	1000
Accuracy	35.562%	43.333%	42.076%	42.324%	43.295%	43.676%

Figure 1: Accuracy in each label

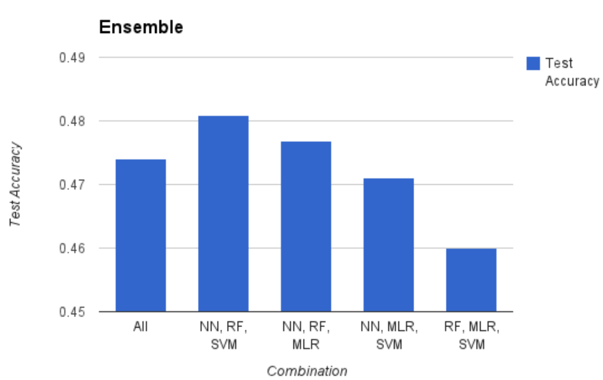


weak labels. For example, although Random Forest seemed have worst performance among 4 classifiers, but it can classify 'airplane' better than other classifiers. SVM is good at 'cat', MLR do better jobs in dog, and Neural Network is good at 'horse' and 'ship'.

The result shows that each classifier has different strong points and weak points. It indicates that if we can find a method to collaborate all classifiers, maybe the aggregated result will be better than single classifier result.

3.6 Ensemble

Figure 2: Ensemble



To aggregated each classifierâĂŹs prediction, we used simple voting to combine prediction results from different classifiers, which means that the ensemble classifier will predict the most frequent predicted label of classifiers. We tried all combinations with more than 3 classifiers, and the result is show above. According to the experiment, it shows that combining predictions of Neural Network, Random Forest, and SVM has best performance.

Table 3: Ensemble v.s. Single Classifier

Method	Test Accuracy	Improvement
NN + RF + SVM	48.10%	—
NN	47.33%	0.76%
SVM	45.85%	2.25%
MLR	45.41%	2.69%
RF	35.03%	13.07%

Comparing the ensemble method and single classifier results, shown in table 4, the ensemble method improves accuracy about 0.76% $\sim$ 13.07%, which indicates that ensemble method can enhance performance of single classifier.

4. CONCLUSIONS

We conducted a machine learning approach to solve multi-class image classification problem.

First, We systematically tuned four well-known machine learning algorithms: Neural Network, Random Forest, Multinomial Logistic Regression, and Support Vector Machine. We found that Neural Network has best performance in single classifier comparison.

Next, we analyzed the prediction results of four classifiers against 10 labels. We found that each classifier has its strong labels and weak labels, which indicates that ensemble method can improve performance.

Furthermore, we conducted voting ensemble method with different combinations. According to the experiment result, combining the prediction results of Neural Network, Random Forest, and Support Vector Machine enhance total performance best, which can increase accuracy 0.76% $\sim$ 13.07%. The result justified that voting ensemble method can help to improve performance.

5. REFERENCES

[1] L. Breiman. Random Forests. *Machine Learning ()*, 45(1):5–32, 2001.

[2] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural Networks, 1989. IJCNN., International Joint Conference on*, pages 593–605, 1989.

[3] D. W. Hosmer and S. Lemeshow. Applied logistic regression. 2000.

[4] D. G. Lowe. Distinctive image features from

scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004.

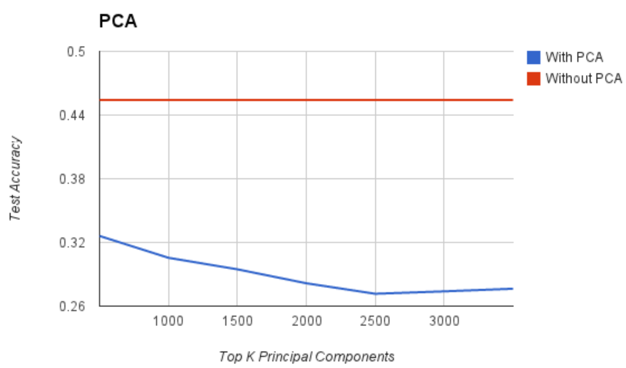
[5] P. Perner. *Machine Learning and Data Mining in Pattern Recognition*. Second International Workshop, MLDM 2001, Leipzig, Germany, July 25-27, 2001. Proceedings. Springer Science & Business Media, July 2001.

[6] V. Vapnik. The Nature of Statistical Learning Theory - Vladimir Vapnik - Google Books. 2000.

APPENDIX

A. DIMENSION REDUCTION

Figure 3: Dimension Reduction



We have tried dimension reduction with PCA, but the result showed that this method did not fit the data.

We used MLR with our basic learning algorithm, and compared the performance of the model trained on raw data, and the model trained on top-K principal components. As the result showed in Figure 3, PCA dimension reduction decreased performance; therefore, we did not use dimension reduction with PCA in model learning process.

B. FEATURE EXPANSION EXPERIMENT

We compared the classifiers trained on raw data and those trained on raw data with expansion features. According the experiments, we found that feature expansion can increase accuracy of all classifiers about 0.2%   3.8% .

The result shows that feature expansion has potential to improve performance. However, because the proposal mentioned that the image feature extraction is not the main point of this project, we did not exploit more sophisticated image features, like SIFT and GIST. Rather, we focus on utilizing different classifiers in this project.

C. DIVISION OF LABOR

SanChuan Hung (sanchuah) Program and tuning Kernel Multinomial Logistic Regression and SVM. Documentation design and test result of these 2 classifiers. Program and test feature expansion, PCA and ensemble method. Label Analysis and result report.

Jiajun Wang (jiajunwa) Learn Weka. Tuning Neural Network and Random Forest classifiers using Weka. Documentation design and test result of these 2 classifiers. Test ensemble method based on pre-processed data set using Neural Network and Random Forest classifiers.

Table 4: Ensemble v.s. Single Classifier

Classifier	Accuracy without expanded features	Accuracy with expanded features
NN	43.543%	47.333%
RF	34.743%	35.029%
MLR	42.971%	45.410%
SVM	42.019%	45.848%

D. WHAT WE LEARNED FROM PROJECT

In this project, we tried several classifiers. Their performance is diverse. For example, although Neural Network appears to predict more accuracy than other classifiers, training and tuning it requires a lot of time and computing resource. So we have the trade-off between the tuning cost and potential accuracy when we choose the classifiers.

Then by combining different classifiers using ensemble methods, we managed to enhance the prediction accuracy. The label analysis shows that different classifiers have specify strengths on certain features/labels. So we think there should be more space for improvement. Due to the limited time and computing resource we have, we must stop at current status. If it is given more resource, we think we could using more classifiers and elaborate our ensemble method to achieve higher accuracy.

Finally, although it is not the emphasis of this project, we find feature expansion based on image processing technology do boost predication accuracy greatly. By using a relatively simple expansion strategy, we improved our test accuracy by more than 4%. We think if we use more complicated strategy, the improvement would be more significant. However, as this is not the priority of this project, we didn't dig too deep in this direction.