EasyChair Preprint

№ 3815

Comparison of ML classifiers for Image Data

Sonika Dahiya, Rohit Tyagi and Nishchal Gaba

July 11, 2020

Comparison of ML classifiers for Image Data

Sonika Dahiya[1], Rohit Tyagi[2], and Nishchal Gaba[3]

[1] Dept. of Computer Science and Engineering,
Delhi Technological University, New Delhi, India
sonika.dahiya11@gmail.com
[2] Dept. of Computer Science and Engineering,
Delhi Technological University, New Delhi, India
rohittyagi295@gmail.com
[3] Co-Founder, Unreal AI Technologies Pvt. Ltd.
New Delhi, India
nishchal@unrealai.xyz

Abstract. In this paper, we have compared performance of various machine learning classifiers such as Multinomial Logistic Regression, Support Vector Machine, Multi Layer Perceptron, Random Forests, Naive Bayes, K Nearest Neighbors, ADA Boost and Convolutional Neural Networks on 2 popular image data sets CIFAR-10 and MNIST. Then we tried to find out reason of performance variance. We also considered the significant of feature extraction and feature selection. Convolutional Neural Networks has been winner over all other ML classifiers. We found out that CNN performs feature extraction and selection automatically which no other classifier is able to do.

Keywords: Artificial Intelligence, Computer Vision, Feature Extraction, Feature Selection, Image Classification, Machine learning

1 Introduction

Machine learning is the quintessential skill of this digital age. As we dissect the process how a machine learns to classify and the inputs or the raw materials needed for learning the specifics of the desired task, features or attributes forms the basis of what we actually feed in the learning algorithm. The collection of data objects, data records, vector data, data tests, data cases or data entity is called data set. Data can be sequential, temporal, sparse, dense, 2D, 3D or high dimensional. As well as data can be represented in tables, graphs, documents, images etc. In this world of digitization, images play a very important role in various areas of life including scientific computing and visual persuasion. Technically images can be binary images, gray scale images , rgb images , hue saturation value or hue saturation lightness images etc. Each data record can be represented via a huge number of features. But all features are not necessarily significant for analysis or classification. Thus feature selection and feature extraction are significant research areas. Feature selection can be defined as a

problem of choosing the minimal set of features that are able to address the problem in a more effective, compact and computationally efficient manner. Feature selection involves creating new features from existing ones, removing redundant and insignificant features, combining a number of features to a minimal count, as well as splitting a feature to a number of features.

Feature extraction involves gathering set of information from the given data and transforming it into smaller number of attributes that carry the maximum information about the original data. For example: Iris Data Set [4], which consists of , 150 instances of 3 classes (Iris Setosa, Iris Versicolour & Iris Virginica) featuring 4 attributes, namely, sepal length (in cm), sepal width (in cm), petal length (in cm) and petal width (in cm), only these 4 attributes are sufficient for the classification of flowers in this dataset. As there can be many features for any one particular flower image like number of leaves, plant cell, length of the stem, plant structure, chloroplast, photosynthesis process, sepal length,sepal width, petal length,petal width etc . So, the transformation of the image to numerical attributes is feature extraction. This allows classifiers to operate on the image data. The selection of only these 4 features (sepal length, sepal width, petal length, petal width)for classification or any further processing is feature selection. In this paper we mainly take up feature extraction and feature selection for image specific tasks, and how the ocean of classifiers learn from the image directly or indirectly (after feature extraction or selection) to segregate them into desired classes. A. Jović et al. majorly classified the feature selection methods into three categories, filter methods, wrapper methods and hybrid methods[6]. It takes on the specific discussion of Image classification, and the challenges to represent an image for the classifier as well. It is important to understand the cycle of image classification given in Fig. 1 to understand the role played by feature extraction.

D.Lu and Q.Weng presents tables for Taxonomy of image classification methods and Major Advanced Classification Methods which give insight into the classifiers performance based on the desired task and type of features[10]. As type of features forms the basis for further type of analysis, feature extraction and feature selection form a strong foundation for image classification. This may be done externally by applying the techniques separately for extraction and selection or automatically using a classifier such as Convolutional Neural Network (CNN), which internally performs this hierarchy, meaning, we dont have complete control over the feature extraction over for each layer, specifically telling the network to extract only a particular feature of choice and then performing different feature selections on it to get the best set of features but rather the weights are updated and features are learnt hierarchically.

2 Pre-Processing the Original Images

In this work we refer to image as the pixel value matrix data for the RGB (Red, Green, Blue) values. The processing stage may involve changing it from the RGB to gray-scale and then performing the feature extraction and selection.

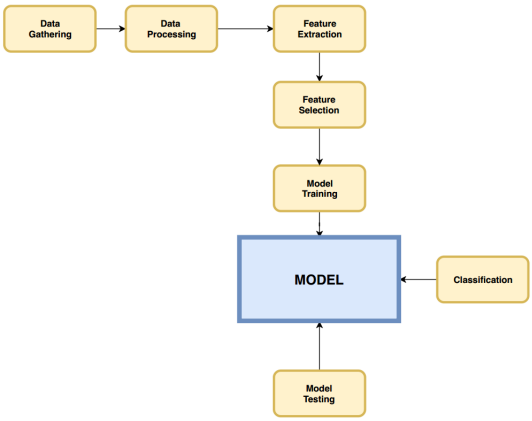

Fig. 1. Cycle for Image Classification Model

The conversion to gray-scale would mean, reduction of the data by 67% as we would drop 2 channels which could technically help to speed up the training time which is a crucial factor sometimes in Deep CNNs which may take upto months to train. Zheng et al. implements a compact CNN [14], achieving almost the similar accuracy as RGB images on CIFAR-10 [7] dataset. This actually performs well as although the RGB matrix does have different values for color information but actually the spatial features are not lost in conversion to a gray-scale matrix. Further improvement may be there using Background/Foreground enhancement [11], that would enable the network to identify boundaries more easily rather than separating the background and foreground features itself.

Although it seems promising, but in real life problems, the gray-scale classification actually is unable to differentiate two similar objects of different colors. Gray-scale is good for tasks where only the shape of the object can enable the classifier to perform well. But such classifiers may perform poorly to differentiate out objects such as red and blue t- shirts from each other. Hence we conclude that when choosing the gray-scale preprocessing, the disadvantages of loosing the color classification should be taken into account. To compensate for this lack of color differentiation, Zheng et. Al, uses a histogram of bins, to store pixel counts for different ranges of bins, somewhat preserving the colored information [14]. This may have more overhead than the original gray-scale network itself, but it does preserve the color information in some magnitude. In case of Foreground enhancement, contours may be drawn over the original images for the object de-

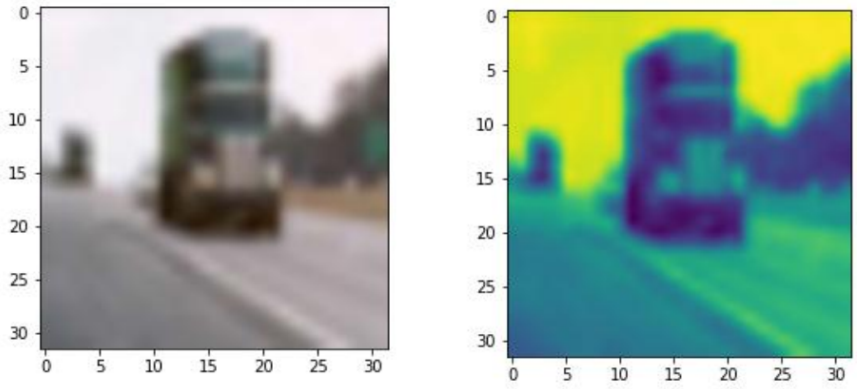

Fig. 2. Gray-scale image conversion using channel drop

tection or temporal difference methods could be used, to make the actual object of insert easier to identify based on the newly created pixel differences. As seen in Fig. 2, we use the single channel technique for gray-scale conversion by dropping the rest of the channels to preserve the spatial information. If the dataset is skewed or biased towards a particular class, most classifiers will just over-fit and give bad results on testing sets usually. Data Augmentation compensates for this to an extent, as it helps increase the size of the dataset and introduce more variation in the data itself as well, which will in turn help the feature extraction algorithms to get a largest set of values. Augmentation techniques involves flipping, translation, scaling and others. Popescu et. al designed 48 features for the dataset of public pollen image dataset [11]. Specific to their task, they took features such as height, width, the dimensions of ellipse enclosing the object. This is totally different for some other form of data. For CIFAR-10 dataset, usually the automated form of CNNs or Deep CNNs is preferred. With huge number of images, it is difficult to calculate features of different objects such as height, width, color information, the contours and creating this set of numerical features and have human verification for it to pass over to the network. For most tasks involving a new dataset, the feature extraction has to be created in a custom manner if not using the automated method. Hence it is important to understand the dataset as it drives the process of feature extraction and it may actually involve more work designing a good feature extractor, rather than the classifier itself.

3 Classifier Details

3.1 Multinomial Logistic Regression (MLR)[9]

MLR classifiers predicts probabilities of binary classes known as logistic regression and if for more than two classes multinomial regression in terms of the dependent output variables. Logistic function predicts the probability for a particular outcome by formation of a linear combination of independent features. Hence, it is a linear classifier. It follows a Bernoulli distribution for dependent variables in case of two classes.[9]

3.2 Support Vector Machine (SVM)[3]

SVM when used for classification tries to find a hyperplane differentiating different classes. It is a linear classifier but can be used as a non-linear classifier using kernel implementation by mapping data to a higher dimensional space called as feature space. There may be many hyperplanes separating the two classes but an optimal hyperplane is defined as the linear decision function which has max margin between the class vectors [3].

3.3 Multi Layer Perceptron (MLP)[13]

MLP is a feedforward neural network using input, output and hidden layers with primarily linear relationship between the layers and the activation functions. A basic neuron consists of the structure,

$$\mathbf{y} = \mathbf{w} \cdot \mathbf{x} + \mathbf{b} \tag{1}$$

Here $\mathbf{y}$ = output of the layer, $\mathbf{w}$ = weights of the hidden layer, $\mathbf{x}$ = input to the hidden layer, $\mathbf{b}$ = bias term to be added for the neurons. MLP uses a back propagation neural network for classification to calculate the loss and optimize the values of w and b to minimize the corresponding error and in turn approximates the Bayes optimal discriminant function.

3.4 Random Forests (RF)[2]

Random Forest uses an ensemble of trees to randomly generate the trees using the training input vector to predict the output vector, similar in analogy to generate a random set of weights, independent of the past weight sequences. Then the best one is voted in and the process is repeated for a fix number of times and the best tree is selected as the corresponding classifier.

3.5 Naive Bayes (NB)[12]

Naïve Bayes uses an approximate Bayesian distribution over the dataset and predicts the most probable class based on the features. Although it assumes, independence of features, which is not always the case, but still in many cases, NB has given competitive results because optimality in terms of classifier error is not always dependent on the quality of approximate distribution.

3.6 K Nearest Neighbors (KNN)[1]

K-NN is a form of non-parametric based classifier, which depends on data neighbors rather than intensive training of the parameters of the network. It classifies the input data based on the class of the nearest item in the dataset using Nearest-Neighbor based distance estimation. Due to this, sometimes over fitting can be avoided and classification can be done faster than parametric learning based classifiers.

3.7 ADA Boost (ADA)[5]

Boosting is used to improve the performance of learning algorithms. It runs small learners over various distributions of the training set and then combine them into a single composite classifier, where associated weights are taken with these classifiers and are updated to improve the training accuracy on different samples. This can sometimes bear huge individual errors in some learners but overall their composite classifier can still give good results, hence it is robust to unstable behavior of the learners.

3.8 Convolutional Neural Networks (CNN)[8]

CNNs have huge learning capacity which makes it great for tasks such as image classification and object recognition. It uses variation of breadth and depth of features to extract features and learn from the data. The learning capacity can be varied by changing the size and number of different layers. Although convolution itself is a linear operation, non-linearity can be added using activation layers. A standard CNN usually consists of other layers such as Dropout, Max Pooling, Batch Normalization and Fully Connected Layers. But it is computationally expensive to train and is good at spatial features compared to temporal features.

4 Simulation And Results

We implemented MLR, SVM, MLP, RF, NB, K-NN, ADA, CNN classifiers using Python 3.6, Jupyter Notebook IDE on Ubuntu 17.04 with 8 GB RAM, Intel i7 (4th Gen Processor).Comparison of classifiers is drawn on 2 popular image datasets CIFAR-10 and MNIST. The CIFAR 10 is a collection of 60,000 RGB images, which belong to 10 classes. The classes of CIFAR 10 images dataset

are of dogs, cats, airplanes, deer, automobiles, birds , frogs, horses, ships and trucks. Each class is having 6,000 images. Each image is of 32*32 pixels with 3 channels for each pixel i.e. red, green and blue. MNIST dataset contains a set of penmanship scanned images of numerals, the numerals scale from 0 to 9. All the images belongs to 10 classes such that the images of digit 0 belongs to class 0, the images of digit 1 belongs to class 1 and so on. Each gray-scale image is 28*28 pixels each. The accuracy results for the classifiers on validation and test data for CIFAR-10 and MNIST are presented in Table 1 using random shuffle and division to choose the 40,000 training, 10,000 validation and 10, 000 testing images. Table 2 presents the cross-validation accuracies and validation set accuracies for MNIST dataset. Cross-validation is not done CNN and SVM due to their high computational cost and the variation of Cross-validation accuracies was observed to be only around 1% in case of other classifiers, hence we skip the cross validation for these two.

Table 1. Classifier Accuracies on CIFAR-10 dataset

S.No.	Classifier	Validation Accuracy	Testing Accuracy
1	Multinomial Logistic Regression	40.64%	40.22%
2	Support Vector Machine	42.55%	41.94%
3	Random Forest Classifier	26.57%	25.74%
4	Multi-Layer Perceptron	43.52%	43.34%
5	K-Nearest Neighbor	31.94%	31.37%
6	Ada Boost	30.67%	30.34%
7	Naïve Bayes	29.44%	28.89%
8	Convolutional Neural Network	80.50%	65.54%

Table 2. Classifier Accuracies on MNIST dataset

S.No.	Classifier	Cross-Validation Accuracy	Validation Accuracy
1	Multinomial Logistic Regression	[91.20%, 92.22%, 91.41%, 91.71%, 92.64%]	92.60%
2	Support Vector Machine	—	94.39%
3	Random Forest Classifier	[62.24%, 63.28%, 63.73%, 62.36%, 66.34%]	64.70%
4	Multi-Layer Perceptron	[94.81%, 95.16%, 95.22%, 93.55%, 95.79%]	95.63%
5	K-Nearest Neighbor	[96.80%, 96.88%, 96.89%, 96.59%, 97.08%]	97.00%
6	Ada Boost	[72.49%, 70.14%, 70.52%, 70.77%, 75.58%]	71.26%
7	Naïve Bayes	[55.04%, 56.02%, 55.13%, 54.65%, 55.89%]	54.92%
8	Convolutional Neural Network	—	98.00%

For Logistic Regression, we used multinomial logistic regression with Limited-memory BFGS (LBFGS) optimizer and l2 penalty. SVM Support Vector Classi-

fier (SVC) tuned with penalty parameter 1.0, rbf kernel and 3 degree polynomial kernel function. Random Forest Classifier is implemented with 100 trees for estimation with a maximum depth of 2 using gini criterion. Multi Layer Perceptron (MLP) classifier is used with a setting of 100 hidden layers, Rectified Linear Unit (ReLU) activation, Adam optimizer and L2 penalty parameter as 1. KNN classifier uses 3 nearest neighbors and uniform weight initialization. Ada Boost Classifier implements 50 estimators with a learning rate of 1.0 using SAMME.R estimator. Multinomial Naive Bayes Classifier done using additive smoothing parameters with enabled calculation of priors on data. CNN used for MNIST consists of 4 layers, 1 convolution layer with kernel size of 3 and 8 output channels, max pooling layer with kernel and stride of 2 each, fully connected layers with 150 and 10 hidden units output each. For CNN used in CIFAR-10 gray-scale network, we center crop the images from (32, 32, 3) to (24, 24, 3) and then use the channel drop technique to keep a single channel to convert the input to (24, 24, 1). The network consists of 4 convolution layers with ReLU activation and 2 of these layers actually work as an inception type model as the filter and stride are set to 1, then 3 normalization layers are used with 2 pool layers along with 2 fully connected layers at the bottom of the model. The results from the table show that CNN outperforms the other classifiers, which is partly due to their high learning capability. Logistic Regression, Naive Bayes Classifier are linear classifiers and hence have limited capability to capture non-linearity between the input data and output classes. Non- linear classifiers, especially CNN with activation functions can capture a higher degree of relationship. For the sake of simplicity, only a single model of CNN is implemented. We test out additional methods to speed up the CNN as their computational cost can rise high. RGB CIFAR-10 images are cropped and converted to gray-scale before passing to the CNN. As CNNs capture spatial information, the accuracy drop is not too high in comparison to the gain in the speed of implementation.

5 Conclusions

In this paper, the performance of various classifiers such as Logistic Regression, SVM, Random Forest, Multi Layer Perceptron, KNN , Ada Boost, Nave Bayes, CNN are tested on CIFAR-10 and MNIST dataset. We train classifiers to map the images into 10 classes using different algorithms. To improve the performance and time taken by various classifiers, significance of choosing right and minimal set of features is very crucial. So, for all classifiers other than CNN different feature selection and feature extraction algorithms are required. But CNN automatically performs feature extraction. This is one of major advantages of using CNN for image classification.

One disadvantage is that CNNs easily scale up to thousands of learn able parameters, making it much more cost intensive. We implemented an additional technique with CNNs in attempt to speed them up using Gray-scale images rather than RGB images in CNN. As CNNs only capture spatial information, the accuracy drop should not be huge, and a balance between accuracy and

speed can be struck. Currently, the input to the model is a 32x32x3 image for CIFAR-10, which makes up 3072 features. This number can be reduced using process called Feature Selection, where we select most contributing features and drop the rest. This technique works well on numerical data and independent features, but in case of image data, random pixels cannot be discarded, as it will lead to huge loss of the data.To improve the performance and time taken by various classifiers, significance of choosing right and minimal set of features is very crucial. So for classifiers other than CNN, good feature extraction and selection will help improve the results substantially. Hence, we conclude relationship in image data is different from independent column numeric data and hence, the preprocessing has to be performed accordingly. CNNs are more adaptable to such image datasets as they can capture higher degree relations with automated feature extraction and selection.

6 Future Work

CNNs outperformed other classifiers such as Logistic Regression, SVM, Random Forest, Multi Layer Perceptron, KNN , Ada Boost, Nave Bayes in task of classification on CIFAR-10 and MNIST dataset for images. Currently, The hyperparameters in these classifiers are manually provided or taken in a reasonable range. CNNs include weight initialization, bias initialization ,learning rate and other layer related parameters to be initialized, setting them appropriately helps in converging to the results faster. There is no definite known algorithm for setting them at a particular value, it is completely data dependent as the patterns in data will vary from one dataset to another. Same goes for the case of SVM and others, although there are methods that can help optimize the parameters of the SVM such as Genetic Based SVM. Although the number of features and their degree of relationship is quite high for linear or quadratic classifiers to capture. Currently, Feature Extraction and Feature Selection techniques are not applied to their full extent, and most of the time, trans formative difference between these is not understood which leads to degraded results. These techniques can help reduce the input to the model and speed it up but they work good mostly on independent features. For pixel matrices, having values varying between 0-255 and arranged in a continuous manner, these techniques does not seem to perform great, and their computational cost of application may make it infeasible for image classification. CNNs performed overwhelmingly for the dataset, but there is no formula to find the best architecture for CNN and initialization of their parameters. For a novice designing the CNNs is a herculean task. It is easier to work with smaller and more familiar classifiers such as SVM. CNNs have lots of variations in terms of layers such as Dropout, Activation, Batch Normalization, making it quite complex and difficult to understand. Hence, in future we will compare various architectures of CNNs and efficient parameter tuning along with its effect on feature extraction, feature selection and speed of optimization for CNN.

References

1. Boiman, Oren, Eli Shechtman, and Michal Irani. "In defense of nearest-neighbor based image classification." Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE, 2008.
2. Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32
3. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.
4. Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, Irvine, CA: University of California, School of Information and Computer Science, Iris Data Set, Fischer 1936
5. Freund, Yoav, Robert Schapire, and Naoki Abe. "A short introduction to boosting." Journal-Japanese Society For Artificial Intelligence 14.771-780 (1999): 1612.
6. Jović, A., Brkić, K., & Bogunović, N. (2015, May). A review of feature selection methods with applications. In Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on (pp. 1200-1205). IEEE.
7. Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.
8. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
9. Kwak, C., & Clayton-Matthews, A. (2002). Multinomial logistic regression. Nursing research, 51(6), 404-410.
10. Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. International journal of Remote sensing, 28(5), 823-870.
11. Popescu, M. C., & Sasu, L. M. (2014, May). Feature extraction, feature selection and machine learning for image classification: A case study. In Optimization of Electrical and Electronic Equipment (OPTIM), 2014 International Conference on (pp. 968-973). IEEE
12. Rish, Irina. "An empirical study of the naive Bayes classifier." IJCAI 2001 workshop on empirical methods in artificial intelligence. Vol. 3. No. 22. New York: IBM, 2001.
13. Ruck, D. W., Rogers, S. K., Kabrisky, M., Oxley, M. E., & Suter, B. W. (1990). The multilayer perceptron as an approximation to a Bayes optimal discriminant function. IEEE Transactions on Neural Networks, 1(4), 296-298.
14. Zheng, Z., Li, Z., Nagar, A., & Kang, W. (2015). Compact deep convolutional neural networks for image classification. ICMEW, 1-6.