International Journal of Advanced Multidisciplinary Research
ISSN: 2393-8870
www.ijarm.com
DOI: 10.22192/ijamr Volume 5, Issue 5 -2018

Research Article

DOI: http://dx.doi.org/10.22192/ijamr.2018.05.05.005

Learning Rate in Supervised Learning Models – The Case of Digit Identification

Andy W. Chen

2053 Main Mall, Vancouver, BC, Canada
*Corresponding Author: *andywchenca99@gmail.com*

Keywords

Machine Learning,
Digit Recognition,
Image Identification,
Supervised Learning

Abstract

In this paper, I explore the application of machine learning models for handwritten digit identification. The goal is to train and predict multiple machine learning models with a dataset of handwritten digits. I vary the learning rate of the model to investigate the relationship between learning rate and prediction accuracy. I find that the prediction accuracies for both training and test data increase between learning rate of 0 and 0.1, and then fluctuate around a constant beyond learning rate of 0.1. The fluctuations are greater for pairs of digits that are more difficult to distinguish.

1.0 Introduction

Digit identification is a task required in a variety of industries. For example, a traffic control systems may use an automated system with machine learning algorithms to identify license plate numbers of vehicles that exceed the speed limit. For another example, an application on a cell phone with touch screen may use digit identification to read the handwritten digits of users in real time. These machine learning algorithms first require training models using labeled data. The data should be a diverse data set covering a large variety of handwritten digits. Training the model requires tuning several parameters such as the learning rate. Testing the model is a way to test the accuracy of the models on new data for evaluation purposes.

Numerous works have been in the area of digit identification. For example, Biglari et al. (2014) proposed a new approach for Persian handwritten digit recognition that uses Local Binary Pattern (LBP) operator to extract features. Shilbayeh et al. (2013) built a model for efficient Hindi digit recognition using multilayer perceptron neural network with backpropagation. Harifi and Aghagolzadeh (2007) proposed the use of neural network to first extract features and then train the model using another neural network. They achieved a high recognition rate of 97.6%. Alani (2017) presented a new approach for Arabic digit recognition with a combination of restricted Boltzmann machine and convolutional neural network (CNN). Shah and Yousaf (2007) created a database of handwritten digits using a new data collection method and train neural networks using the model.

This paper presents an analysis of machine learning models used for digit identification. In particular, I train and predict multiple models using a large dataset of handwritten digits. I tune the learning rate, a hyper-parameter, and compare these models to explore the effect of learning rate on the accuracy of the predictions. The machine learning model I use is the binary logistic regression.

2.0 Methods

I use the MNIST (Modified National Institute of Standards and Technology) dataset, which is a large collection of handwritten digits. Each digit is stored in a vector of 784 binary values, where each element represents a part of an image with size 28 by 28. Each vector also contains a 785thelement representing the class of the data point (the value of the digit in this case). The MNIST data set contains 60,000 training samples and 10,000 test samples across the ten possible digits from 0 to 9.

To explore the effect of learning rate on the accuracy of the logistic regression model, first I show the derivation of the algorithm and how the learning rate is used in the algorithm for predicting the digit. The algorithm is called gradient descent.

I build binary logistic regressions, where the target variable y is the value of the digit, x is the set of features (the 784 binary values in the vector that represents a digit), and θ is the set of parameters to be estimated. θ is also a vector of 784 values, where each value is the weight given to one of the 784 elements of the image with size 28 by 28.

First I set the probability of having a label $y = 1$ be $P(y = 1|x, \theta) = \sigma(x)$ and $y = 0$ be $P(y = 0|x, \theta) = 1 - \sigma(x)$, where $\sigma(x) = \frac{1}{1+e^{-\theta^T x}}$ is the sigmoid function. The sigmoid function is used for transforming the weighted average of the features into a numerical value between 0 and 1 that can be used to predict the class of the target value.

Suppose there are$N$ data points. The joint probabilities of all the labels y is

$$p(y|x) = \prod_{i=1}^{N} \left(P(y^i = 1|x^i)\right)^{y^i} \left(1 - P(y^i = 1|x^i)\right)^{1-y^i} = \prod_{i=1}^{N} \left(\sigma(x^i)\right)^{y^i} \left(1 - \sigma(x^i)\right)^{1-y^i}$$

This is the same equation for the likelihood of the parametersθ, so I write

$$l(\theta) = \prod_{i=1}^{N} \left(\sigma(x^i)\right)^{y^i} \left(1 - \sigma(x^i)\right)^{1-y^i}$$

Taking the log of the likelihood $l(\theta)$ yields

$$L(\theta) = \log(l(\theta)) = \sum_{i=1}^{N} y^i \log\left(\sigma(x^i)\right) + (1 - y^i)\log\left(1 - \sigma(x^i)\right)$$

The loss function is a function to minimize, which is the opposite of maximizing the log-likelihood function. Therefore, the loss function can be written as

$$-L(\theta) = -\log(l(\theta)) = -\sum_{i=1}^{N} y^i \log\left(\sigma(x^i)\right) + (1 - y^i)\log\left(1 - \sigma(x^i)\right)$$

To derive the gradient of the loss function with respect to model parameters, I take the first derivative of $L(\theta)$ with respect to each parameter θ_j in the set of featuresθ.

$$\frac{\partial L(\theta_j)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} \sum_{i=1}^{N} y^i \log\left(\sigma(x^i)\right) + (1 - y^i)\log\left(1 - \sigma(x^i)\right)$$

$$= \sum_{i=1}^{N} \left(\frac{y^i}{\sigma(x^i)} - \frac{1 - y^i}{1 - \sigma(x^i)}\right) \frac{\partial}{\partial \theta_j} \sigma(x^i)$$

$$= \sum_{i=1}^{N} \left(\frac{y^i}{\sigma(x^i)} - \frac{1 - y^i}{1 - \sigma(x^i)}\right) \sigma(x^i)\left(1 - \sigma(x^i)\right) x_j^i$$

$$= \sum_{i=1}^{N} \left(\frac{y^i - \sigma(x^i)}{\sigma(x^i)(1 - \sigma(x^i))}\right) \sigma(x^i)\left(1 - c(x^i)\right) x_j^i$$

$$= \sum_{i=1}^{N} \left(y^i - \sigma(x^i)\right) x_j^i$$

To speed up training of the model, I use one sample at time to update the parameters, so the gradient can be written without the summation as

$$\frac{\partial L(\theta_j)}{\partial \theta_j} = \left(y^i - \sigma(x^i)\right) x_j^i$$

The equation for updating each parameter θ_j is

$$\theta_j' = \theta_j - \alpha \frac{\partial L(\theta_j)}{\partial \theta_j}$$

where α is the learning rate between 0 and 1.

The idea of the gradient descent is to repeat the estimation of the iterations. Each iteration will update each parameter θ_j for each feature. The algorithm should up when the difference between subsequently estimated parameters θ_j and θ_j' are within a specified threshold, usually a very small number such as 0.0001.

The learning rate α determines how fast the algorithm will converge. However, there are disadvantages of using a learning rate too small or large. If the learning rate is too small, it may take a long time for the model to train. If the learning rate is too large, it may cause the subsequent estimated parameter to decrease by too much and skip the optimal value (usually a local minimum). Therefore, the purpose of the paper is to explore the effect of the learning rate on accuracy of the model.

3.0 Results

I train and test each binary logistic regression model with a different learning rate from 0 to 1 with an increment of 0.1. I use a subset of the MNIST data. In particular, I use the digits 0, 1, 3, and 5. I train and test models to predict 0 or 1. I train and test another model to predict 3 or 5. In total I run 20 models. The results are shown in the graphs below. Figure 1 shows how the accuracy of training and test sets for digits 0 and 1 change with respect to learning rate. Figure 2 shows how the accuracy of training and test sets for digits 3 and 5 change with respect to learning rate. The accuracy of distinguishing 0 and 1 is between 0.96 and 1 for all learning rates, which is higher than the accuracy of distinguishing 3 and 5 (between 0.8 and 0.93). There does not seem to be large drop offs in accuracy as learning rate increases for both models.

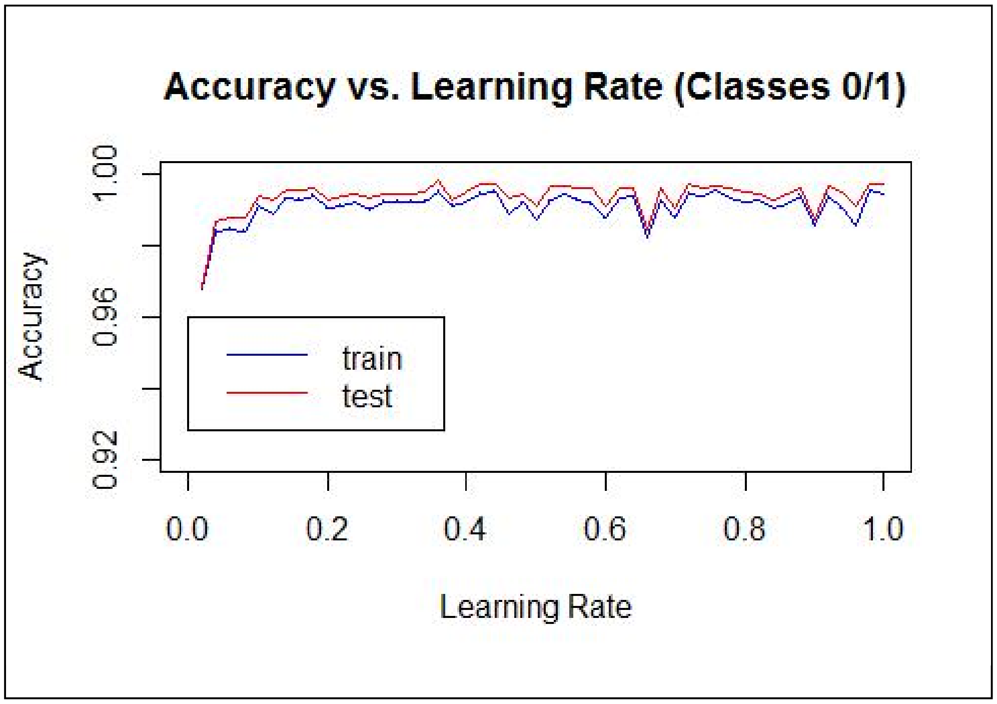

Figure 1. Training and test accuracies vs. learning rate (digits 0 and 1).

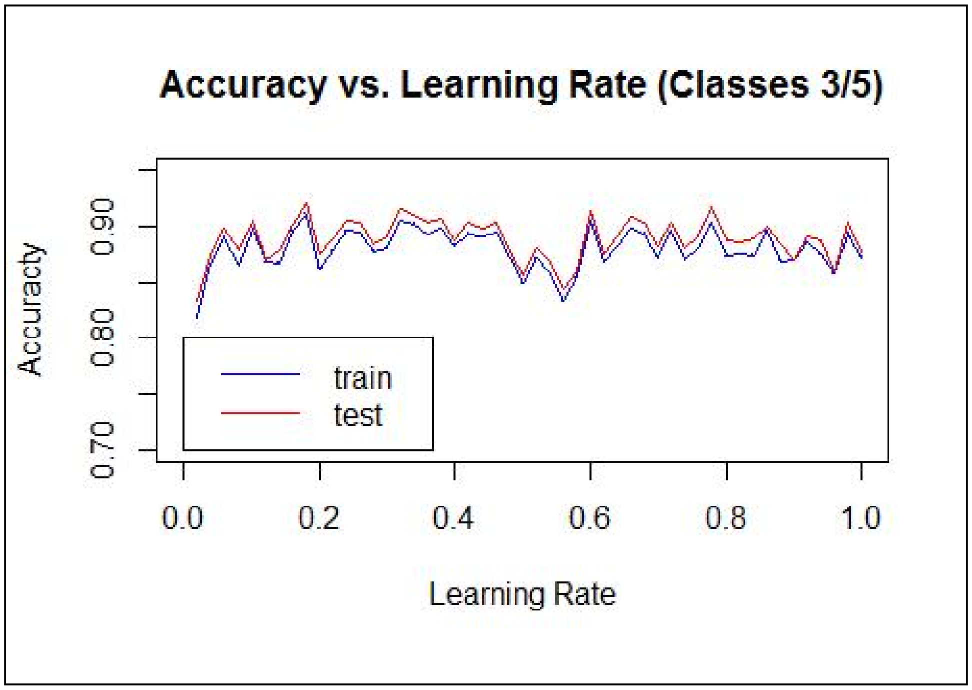

Figure 2. Training and test accuracies vs. learning rate (digits 3 and 5).

4.0 Discussion

Both models show a similar trend. The accuracies have an increasing trend between learning rate of 0 and 0.1. The trend becomes fairly consistent beyond learning rate of 0.1, fluctuating around a constant. The fluctuations are greater for the pair 3 and 5 than for 0 and 1.

Both models show high accuracies for training and test sets. The model for predicting 0 and 1 has higher accuracy for the one predicting 3 and 5. The reason could be that it is easier to distinguish between 0 and 1 than 3 and 5. 0 and 1 have more distinctive features relative to each other. 0 has a round shape, while 1 is a single, straight stroke. On the other hand, 3 and 5 both have a curly shape. The bottom half of each digit is virtually the same. Only the top half varies to a small extent.

The high accuracies across all learning rate could also be due to the comprehensiveness of the MNIST dataset. The MNIST data contains a large number of handwritten digits, so using it to train models to provide a great basis that covers a large variety of handwritten digits. This allows the model to be robust to new data and still be able to achieve high accuracies.

5.0 Conclusion

This project shows the results of binary logistic regression, a supervised machine learning model for predicting handwritten digits. In addition, this project explores the effect of learning rate on the training and test accuracies of machine learning models. For large data sets that cover a comprehensive set of possible data values, the learning rate tends to have less impact on the accuracies. The robustness of the model mainly comes from the excellent training data used. Future extensions may include testing the effect of learning rate on other machine learning models such as support vector machines and decision trees. The results can be used to implement automatic processes for digit identification in businesses such as mobile phones and cameras. The applications include identifying handwriting from users on mobile phones and license plate numbers on the roads.

References

Alani AA. 2017. Arabic handwritten digit recognition based on restricted Boltzmann machine and convolutional neural networks. 8(4):142.

Biglari M, Mirzaei F, Neycharan JG. 2014. Perisan/Arabic handwritten digit recognition using local binary pattern. International Journal of Digital Information and Wireless Communications (IJDIWC). 486-492.

Harifi A, Aghagolzadeh A. 2007. A new pattern for handwritten Persian/Arabic digit recognition. International Journal of Computer and Information Engineering. 1(3):798-801.

Shah FT, Yousaf K. 2007. Handwritten digit recognition using image processing and neural networks. Proceedings of the World Congress on Engineering.

Shilbayeh NF, Alwakeel MM, Naser MM. 2013. An efficient neural network for recognizing gestural Hindi digits. American Journal of Applied Sciences. 0(9):938-951.

Access this Article in Online	
Quick Response Code	Website: www.ijarm.com
	Subject: Information Technology
DOI:10.22192/ijamr.2018.05.05.005	

How to cite this article:
Andy W. Chen. (2018). Learning Rate in Supervised Learning Models – The Case of Digit Identification. Int. J. Adv. Multidiscip. Res. 5(5): 38-42
DOI: http://dx.doi.org/10.22192/ijamr.2018.05.05.005