Number Recognition via thresholded grayscale value '74'

Junyuan Fang

April 1, 2022

1 Introduction

This report discusses different attempts to classify images into digits. Digit recognition is a simple task for a human, however, in some extreme cases, random handwriting makes it impossible for us to classify numbers, and then we need the help of machine learning models. With the reduced grayscale value, machine learning models classify digit images into 0-9. First of all, section 2 discusses the problem formulation by clarifying the classification task's data points, features and labels. Secondly, section 3 discusses the dataset, feature selection policy and two different models. Both of them are multi-layer perception methods but use different activation functions. Eventually, Section 4 discusses training, validation and test error and the conclusion.

2 Problem Formulation

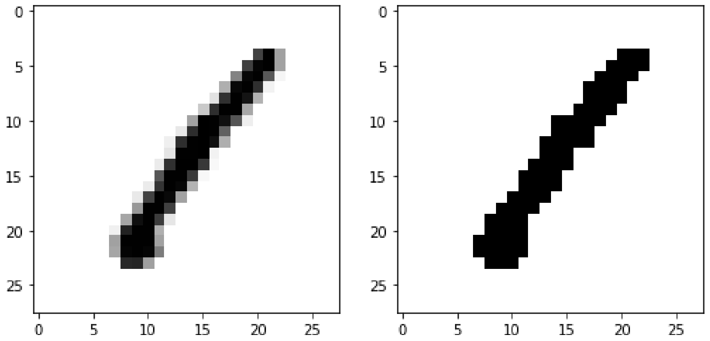
The data points are handwritten digits [1], from 0 through 9. Each example is 28 times 28 grayscale images, in which the pixel is an integer between 0 and 255. The higher value in the pixel means that the pixel is marked more clearly. Original images [1] are from the National Institute of Standards and Technology database (MNIST). MNIST's handwritten digits are collected among Census Bureau employees and high-school students. However, data files in the CSV format [2] are more friendly for applying the machine learning methods.

Labels are integers from 0 to 9. Features are 784 pixels in total. Pixel's value is 0 or 1 by converting grayscale value to binary value through threshold grayscale value 74. Our goal is to achieve digit recognition via threshold grayscale value 74.

3 Methods

3.1 Datasets

The dataset is fetched from the kaggle page [2]. Each data point in the given dataset is consists of digit and digit pixels. In total, there are 42000 data points in the dataset. Pandas package [3] is used for data preprocessing and splitting data points into labels and features. The 'label' column of the dataset is used as the label. After preprocessing



(a) 0-255

(b) 0-1

Figure 1: Original image(left), after threshold grayscale value 74(right)

with the grayscale value 74, ‘pixel 0’-‘pixel 783 ’ columns in the dataset are used as features.

The benefits of performing feature selection before modelling are reducing the overfitting, improving the accuracy and reducing the training time. After applying the feature selection method with the ”least absolute shrinkage and selection operator”(LASSO), features are reduced from 784 to 689. Training time is slightly improved, but the accuracy is also decreased at the same time. LASSO does not give a better outcome. Thus, in the 28x28 picture, all 784 features have equal importance. The process of feature selection is leaving the feature as it is.

For avoiding overfitting, either a smaller hypothesis space or more data points are needed. Thus, a huge part of the data points is split into the training set. The validation set is for testing the model’s performance during the training, and test data are data that the model is never seen before. The validation set and test set can be smaller. The data points are split into training, validation and test sets as 70%, 15% and 15% of the data points.

However, because of the possibility of sequential digits could be written by the same person, data points are shuffled before applying them to the models.

3.2 Multi-layer perception models

Multi-layer perceptron (MLP) is a class of feedforward artificial neural network (ANN) [4]. The reason for applying the MLP method is that it is less sensitive to model complexity compared to regressions, it can approximate any continuous function [5], and it gives robust results after learning the data from the training set. Thus according to the benefits of this model, two different activation functions are applied to solving the digit recognition problem. These different activation functions are sigmoid and ReLU functions.

Multi-layer perceptron (MLP) has at least 3 layers, the first layer is the input layer, and the last one is called the output layer. Our input layer consists of 784 different nodes with a pixel value. The output layer consists of 10 different nodes, each node represents 0-9 different digits. Rest layers are hidden layers. Each node (neuron) in the hidden layer receives values x_j from the previous layer. MLP is fully connected layers, first of all, each neuron computes a weighted sum $z = \sum_j w_j x_j$ of the inputs.

After that, each neuron passes its calculated z to its own activation function $\sigma(z)$. Eventually, the outcome from the $\sigma(z)_i$ in the current layer is represented by the next layer's income x_j. So on and so on ... This step-by-step computation with each layer is so-called forward propagation.

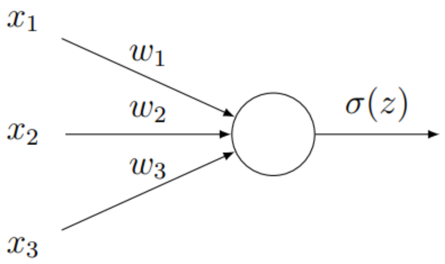


Figure 2: Artificial neural network (ANN) consisting of a single neuron that implements a weighted summation $z = \sum_j w_j x_j$ of its input x_j followed by applying a non-linear activation function $\sigma(z)$. From MLBasicsBook[6]

Different activation functions 'sigmoid' and 'ReLU' are applied to the MLP models. Both of them are a non-linear function

3.2.1 with Sigmoid function

sigmoid function as a activation function.

$$\sigma(z) = \frac{1}{1 + e^z} \tag{1}$$

The activation function (1) is differentiable. Values are between 0 to 1, therefore, the machine learning model can predict the probability as an output.

3.2.2 with ReLU function

Rectified Linear Units (ReLU) as an activation function.

$$\sigma(z) = max(0, z) \tag{2}$$

The advantage of using the ReLU activation function is sparsity [7]. which in concise models that often have better predictive power and less overfitting or noise [7]. ReLU returns 0 if it receives any negative input, but for any positive value x, ReLU returns the positive value x. ReLU is differentiable in the defined area.

3.3 Loss function

The method in our MLP model for minimizing the solution's loss function is called backpropagation. Backpropagation uses a mathematical optimization method e.g. gradient descent to adjust the weight values of each neuron.

For describing the model's performance, we need to define a loss function. In the last layer, we need to compare the label with the output. In this case, the observed data point is label y, and parameter vector $\overrightarrow{w}$ is unknown. We want to use label y to estimate the parameter vector $\overrightarrow{w}$. The problem turns to the maximum likelihood $p(y|\overrightarrow{w})$ estimation problem.

$$f(x) = \frac{1}{1+e^{-x}} \tag{3}$$

Function (3) logistic function can depict the likelihood well [8]. First of all, its value is between 1 and 0. Secondly, its range is the same as probability's range. Thirdly, it is a value either close to 1 or 0. Eventually, it is differentiable.

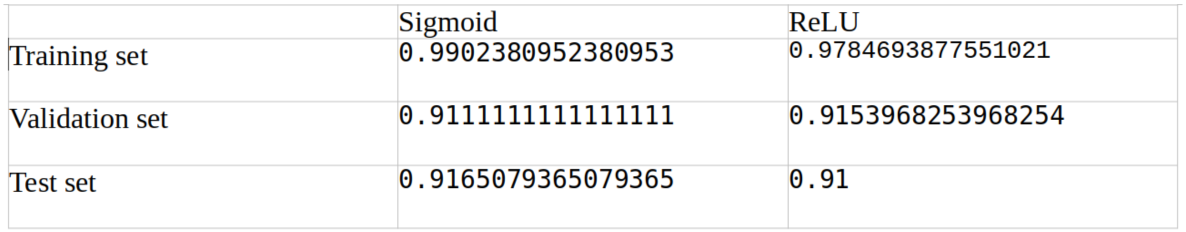
According to the 'MLBasicsBook', Maximum likelihood methods use the loss function[6]

$$L(y, h^{(\overrightarrow{w})}) = -log(p(y|\overrightarrow{w})) \tag{4}$$

When we maximize the likelihood, we are also decreasing the loss function's value. Logistic loss is composed of function (3) and function (4).

4 Result

There are 2 hidden layers in each model, and each layers has 16 neurons. The only difference between two models is the activation function.



	Sigmoid	ReLU
Training set	0.9902380952380953	0.9784693877551021
Validation set	0.9111111111111111	0.9153968253968254
Test set	0.9165079365079365	0.91

Figure 3: Training, validation and test errors

To choose between the two models, I compared their validation errors. MLP with the activation function ReLU has a little bit higher correctness in the validation set than the Sigmoid. Therefore the final chosen method is MLP with the activation function ReLU.

The test set is constructed from 15% of the original data points. However, the test error with the ReLU activation function is 0.09, which is higher than the test error with MLP with the activation function Sigmoid 0.08.

Both models give good results in digit recognition via thresholded grayscale value '74'. Still, the ReLU might be a better selection. Because when the training set is 700, ReLU gives a much better result than the Sigmoid. We are not satisfied with 0.91 correctness. There is still a lot of space for improvement. The 'MLPClassifier' method from sklearn uses the logistic loss, there are different loss functions like mean squared error. As promising directions for future work, we consider using different loss functions, different activation functions or applying a convectional neural network to solve the same problem.

References

[1] The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/. Accessed: 2022-01-25.

[2] Digit-recognizer data. https://www.kaggle.com/c/digit-recognizer/data/. Accessed: 2022-01-25.

[3] Pandas reference. https://pandas.pydata.org/docs/reference/frame.html. Accessed: 2022-03-01.

[4] Wikipedia. Multilayer perceptron — Wikipedia, the free encyclopedia, 2022.

[5] Michael Nielsen. *Neural Networks and Deep Learning*. Determination Press, 1986.

[6] Alexander Jung. *Machine Learning: The Basics*. 2022.

[7] What is rectified linear unit (relu)? — introduction to relu activation function. https://www.mygreatlearning.com/blog/relu-activation-function/#:~:text=ReLU. Accessed: 2022-03-30.

[8] Understanding sigmoid, logistic, softmax functions, and cross-entropy loss (log loss) in classification problems. https://towardsdatascience.com/understanding-sigmoid-logistic-softmax-functions-and-cross-entropy-loss-log-loss-d note = Accessed: 2022-03-30.

Code: https://github.com/junyuan-fang/MachineLearningD_Project