Pattern Recognition with Single Layer Neural Network

Oriol Domingo Roig, Roger Creus Castanyer

April 2019

1 Introduction

In this work we aim to classify set of numbers using a **Data Mining** technique, such as **Neural Networks**. We are going to use a Single Layer Neural Network that will be trained on pictures of size 7x5 pixels per image. Thus, Layer will consist of 35 neurons, each one representing a pixel. Consequently, we will have 35 weights that links a final node with the previous nodes and assign a probability value. This value represents whether the number is in the target number set or not.

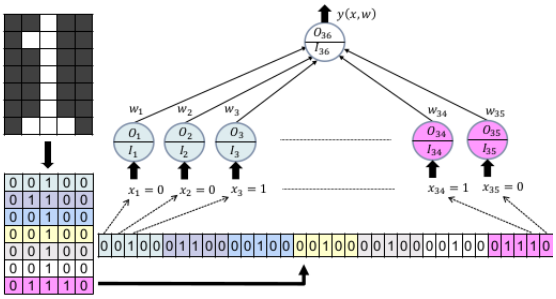


Figure 1: Training Model [1]

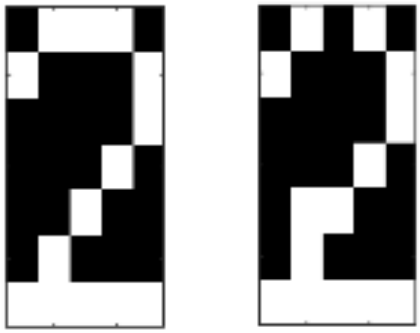
2 Preliminary

This project is divided into three components which ease the understanding of global work flow:

2.1 Data Generation

Training Data and Test Data is generated from a file *"dataSets"* which requires two seeds, number of samples in the training data, target number data set,

frequency of target in data, image noise frequency. It is necessary to specify seeds in order to allow users to recompute model parameters under the same conditions. Moreover, providing the number of samples in the training data is enough, since test data will be built using ten times the amount of training data. Finally, image noise frequency is a mechanism to add some noise in an image, i.e. changing some bits from 0 to 1 and vice versa, then not only is it harder for the model to predict, but also makes it more realistic.



(a) Without noise (b) With noise

Figure 2: Sample

Gathering all this information, *dataSets* is capable of building training and test data under the specific conditions that we imposed. Such script returns four parameters, the first and the third correspond to a matrix in which each row represents 1 pixel out of 35, and each column represent a whole number. Hence, all matrix have the same amount of rows, 35, and differ in terms of columns, number of samples. Then, second and fourth parameters correspond to a vector of Boolean values representing whether such sample belongs to the target number set or not. Finally, first and second parameter is used for training and third and fourth for test.

2.2 Neural Network Functions

The goal of our Neural Network is to train a model in such a way that it is highly accurate when classifying unseen examples. Thus, this model must be built considering the error at each prediction. Our model works with an input, in this case will be an image of 35 pixels with a 0 or 1 value, representing each one a neuron. Then, some weights are applied to each neuron and link all of them to a single value, last neuron, the output. This value is a probability representing whether the image correspond to the target number set or not, as shown in Figure 1.

This result into a loss function that works with some weights, which play a non-linear role respect to input values, and consider predicted values. Moreover, we should apply a common technique in mathematics, known as **L2 regularization**, in order to penalize models that tend to over-fit training data.

$$L(X, y, \lambda) = min_{w \in R^n} L(w; X, y, \lambda) = \sum_{j=1}^{p} (y(x_j, w) - y_j)^2 + \lambda \frac{||w||^2}{2} \quad (1)$$

$$y(x, w) = (1 + e^{-\sum_{i=1}^{n} w_i (1 + e^{-x_i})^{-1}})^{-1} \quad (2)$$

Hyper-parameter definition:

- X_{np} = Training data matrix
- y_p = Target vector

Parameter definition:

- w_n = Weight vector

2.3 Mathematical Optimization Algorithm

3 Behavior and Convergence Analysis

To develop

References

[1] F.-Javier Heredia. Unconstrained optimization. University Lecture, 2019.