Universidad Nacional de Ingenieria

National Peruvian University

Prediction of Numbers

by Logistic Regression and Networks Neuronal

Instructor: Pacheco Taboada Andre

November 1, 2023

Abstract

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaa

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaa

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Contents

1 Introduction **4**

2 Objectives **5**

3 Theoretical Framework **6**

3.1 Machine Learning Fundamentals . . . 6

3.2 Multinomial Logistic Regression . . . 6

3.3 Neural Networks . . . 6

3.3.1 Architecture . . . 7

3.3.2 Backpropagation and Gradient Descent . . . 7

3.4 Convolutional Neural Networks (CNNs) . . . 7

3.4.1 CNN Architecture . . . 8

3.4.2 Feature Learning in CNNs . . . 8

3.5 Image Processing in Machine Learning . . . 8

4 Methodology **9**

4.1 Data Preprocessing . . . 9

4.2 Development of Multinomial Logistic Regression Model . . . 9

4.3 Development of Neural Network . . . 10

4.3.1 CNN Architecture Design . . . 10

4.3.2 Training and Tuning . . . 10

4.4 Integration with Flask . . . 10

4.4.1 Routing and Request Handling . . . 10

4.5 Model Evaluation and Validation . . . 10

5 Resultados **12**

6 Discusiones **12**

7 Appendice **14**

7.1 Regresión Logística Multinomial . . . 14

7.1.1 Fórmula general . . . 14

7.1.2 Función de costo 14
7.1.3 One-Hot Encoding 14
7.2 Redes Neuronales 14
7.2.1 Relación de variables 15
7.2.2 Activación 15
7.2.3 Función de costo 15
7.2.4 Gradiente Descendente 15

1 Introduction

The task of digit recognition, which is a subset of Optical Character Recognition (OCR), has been a significant topic of interest within the Machine Learning community. The ability to accurately identify handwritten or printed digits from images can greatly contribute to various automation systems and is essential in fields such as banking, postal services, and document management systems. This project aims to develop a robust digit recognition system through the application of Multinomial Logistic Regression and Neural Networks, capable of processing images containing digits and predicting the respective numeric values with high accuracy.

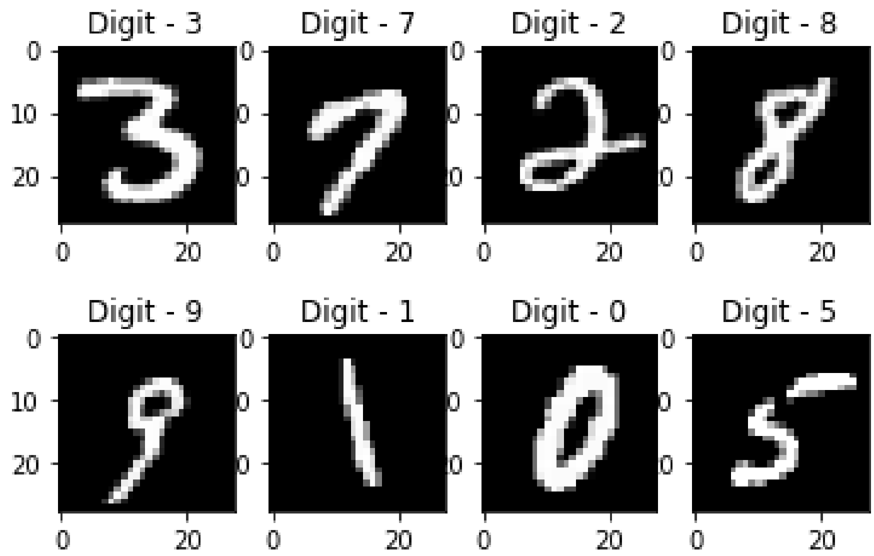

Figure 1: Plot of a Subset of Images From the MNIST Dataset

To achieve this, the project employs Flask, a lightweight WSGI web application framework, for developing a user-friendly interface where users can upload images of digits or manually input digits for prediction. Python, a versatile and powerful programming language, is used for developing the Machine Learning models and handling data processing tasks. The following sections will delve into the theoretical basis and methodologies adopted in this project, focusing on the use of Multinomial Logistic Regression and Neural Networks to meet the stated objectives.

2 Objectives

The primary objectives of this project are outlined as follows:

1. To develop a Multinomial Logistic Regression model capable of predicting digits from images with a high degree of accuracy.

2. To design and train a Neural Network to further enhance prediction accuracy and compare its performance with the Multinomial Logistic Regression model.

3. To implement a user-friendly web interface using Flask, enabling users to upload digit images or input digits manually for prediction.

4. To rigorously evaluate the performance of the developed models, ensuring they meet the required standards for practical deployment.

3 Theoretical Framework

The theoretical foundation of this project lies in the principles of Machine Learning, particularly in Logistic Regression, Neural Networks, and more specifically, Convolutional Neural Networks (CNNs). Understanding these algorithms is crucial for developing a predictive model capable of recognizing digits from images.

3.1 Machine Learning Fundamentals

Machine Learning, a subset of Artificial Intelligence, primarily focuses on the development of algorithms that can learn from and make predictions or decisions based on data. This field encompasses a variety of techniques ranging from simple linear regression to complex deep learning models.[1]

3.2 Multinomial Logistic Regression

Multinomial Logistic Regression, an extension of logistic regression, is used for multi-class classification problems. Unlike binary logistic regression, it can handle scenarios where the outcome can belong to three or more categories.[2]

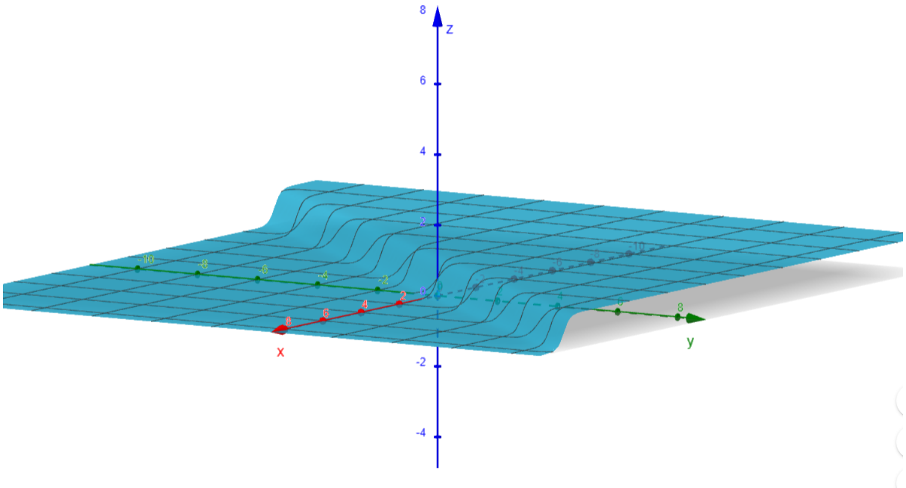

Figure 2: Graphical Representation of Multinomial Logistic Regression

3.3 Neural Networks

Neural Networks are computational models inspired by the human brain's structure and function. They consist of layers of interconnected nodes or neurons and are adept at learning complex patterns and relationships in data.[3]

3.3.1 Architecture

The architecture of a basic neural network comprises input, hidden, and output layers. The input layer receives the data, the hidden layers process the data, and the output layer produces the prediction.

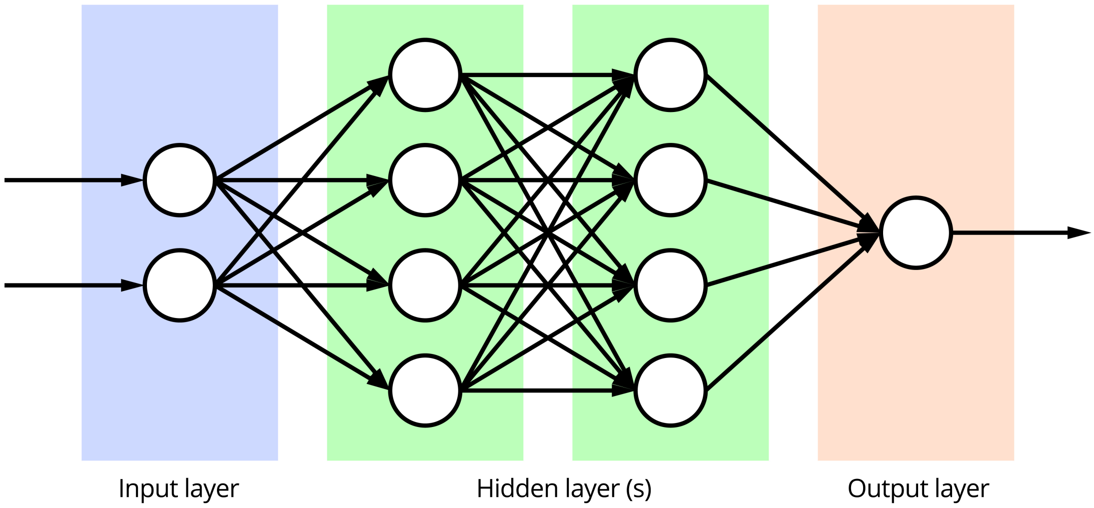

Figure 3: Basic Architecture of a Neural Network

3.3.2 Backpropagation and Gradient Descent

Backpropagation is a method used for computing gradients of the loss function with respect to the weights of the network, essential for the gradient descent optimization algorithm.

3.4 Convolutional Neural Networks (CNNs)

CNNs are a class of deep neural networks, most commonly applied to analyzing visual imagery. They are particularly well-suited for image recognition and classification tasks due to their ability to capture spatial hierarchies in the data.[4]

3.4.1 CNN Architecture

A typical CNN architecture consists of convolutional layers, pooling layers, and fully connected layers. The convolutional layers apply a series of learnable filters to the input, capturing spatial features.

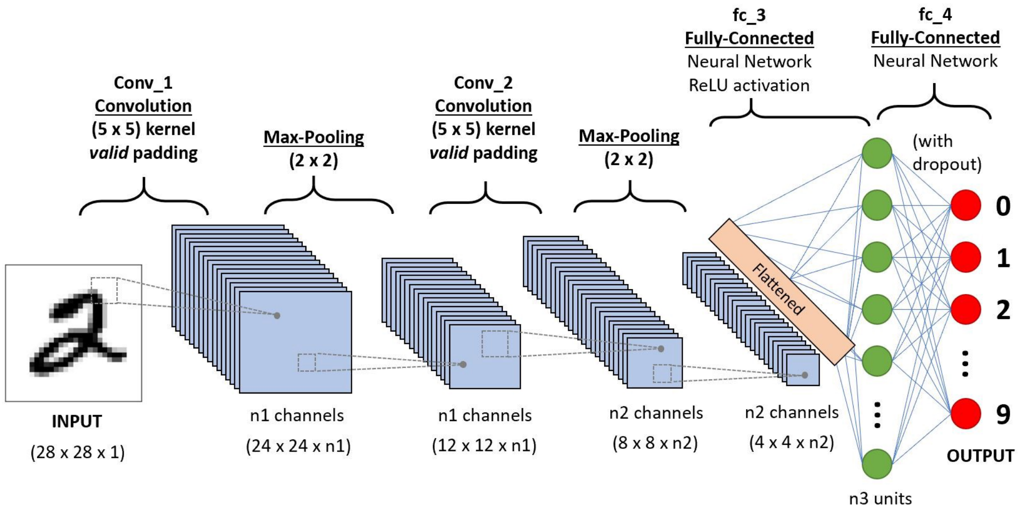

Figure 4: Typical Architecture of a Convolutional Neural Network
Fuente: Di Guan. (2020). Classical Architectures in CNN. Editor 1.
`https://guandi1995.github.io/Classical-CNN-architecture/`

3.4.2 Feature Learning in CNNs

CNNs automatically and adaptively learn spatial hierarchies of features from input images. This feature learning aspect makes CNNs particularly effective for image processing tasks.

3.5 Image Processing in Machine Learning

Processing images to be fed into machine learning models involves several steps like resizing, converting to grayscale, binarization, and transforming them into a matrix or vector format.[5]

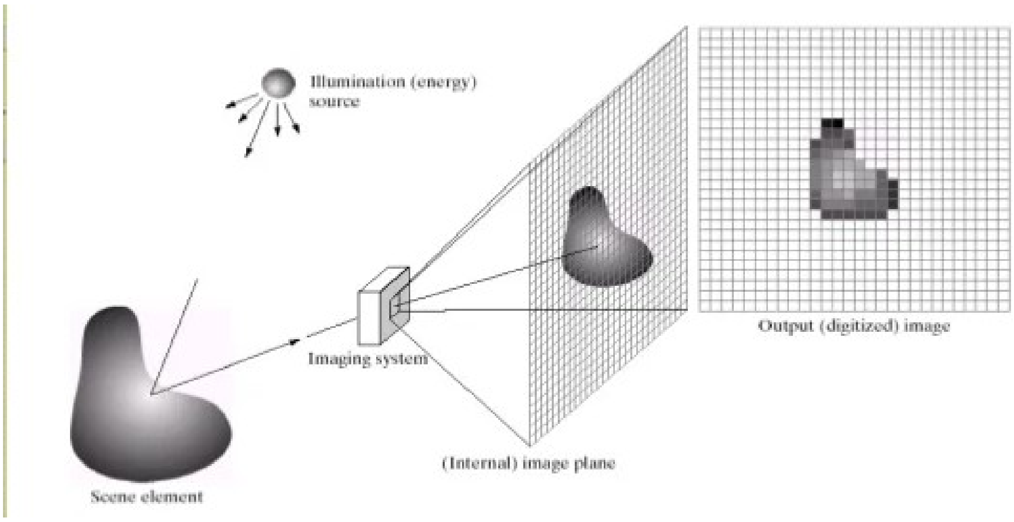

Figure 5: Example of Image Processing Steps
Fuente: Silicon ,M. (2016). Computer Science Thesis in Digital Image Processing. Editor 1. `https://es.slideshare.net/SiliconMentor/computer-science-thesis-in-digital-image-processing#5`

4 Methodology

This section outlines the systematic approach adopted in this project to achieve the objectives delineated earlier, focusing on the development and integration of Multinomial Logistic Regression and Convolutional Neural Networks (CNNs).

4.1 Data Preprocessing

Data preprocessing is a critical step in ensuring that the machine learning models receive high-quality, relevant data for training and prediction.

- **Image Conversion to Grayscale and Binarization:** Images are first converted to grayscale and then binarized to reduce complexity and focus on the essential features.
- **Image Transformation:** The preprocessed images are transformed into a matrix or vector format, making them suitable for feeding into the machine learning models.

4.2 Development of Multinomial Logistic Regression Model

The Multinomial Logistic Regression model is developed to handle the multi-class classification task of digit recognition.

- **Model Training:** The model is trained using a labeled dataset of digit images, where it learns to associate specific features of the images with their corresponding digit labels.

- **Parameter Optimization:** Various parameters, such as the regularization strength and the solver used, are fine-tuned to optimize the model's performance.

4.3 Development of Neural Network

A Neural Network, particularly a CNN, is developed for more advanced digit recognition tasks.

4.3.1 CNN Architecture Design

A CNN is designed with convolutional layers, pooling layers, and fully connected layers tailored to identify complex patterns in the digit images.

4.3.2 Training and Tuning

The CNN is trained on a substantial dataset, with parameters like the learning rate, number of filters in convolutional layers, and number of neurons in fully connected layers being meticulously tuned.

4.4 Integration with Flask

Flask, a Python web framework, is used to create a user-friendly web interface for digit recognition.

4.4.1 Routing and Request Handling

Routing functions are established to handle various web requests, and mechanisms are implemented to process uploaded images or manually entered digits, passing them to the models for prediction.

4.5 Model Evaluation and Validation

Comprehensive evaluation of both models is conducted using metrics such as accuracy, precision, recall, and the F1 Score.

- **Performance Comparison:** The performance of the Multinomial Logistic Regression model and the CNN is compared to determine their effectiveness in digit recognition.

- **Validation Techniques:** Techniques like cross-validation and confusion matrices are employed to validate the models and ensure their robustness and reliability.

5 Resultados

6 Discusiones

References

[1] Hastie, T.; Tibshirani, R.; Friedman, J. (2009). *The Elements of Statistical Learning*. Springer. ISBN 978-0-387-84857-0.

[2] Agresti, A. (2002). *Categorical Data Analysis*. Wiley-Interscience. ISBN 0-471-36093-7.

[3] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT press Cambridge.

[4] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

[5] Gonzalez, R.C., Woods, R.E. (2008). *Digital Image Processing* (3rd ed.). Prentice Hall. ISBN 978-0-13-168728-8.

7 Appendice

7.1 Regresión Logística Multinomial

La regresión logística multinomial es una extensión de la regresión logística binomial para manejar múltiples clases.

7.1.1 Fórmula general

La probabilidad de que una instancia pertenezca a la clase i se da por:

$$p_i = \frac{e^{z_i}}{\sum_{i=1}^{C} e^{z_i}} \tag{1}$$

7.1.2 Función de costo

La función de costo para la regresión logística multinomial es la entropía cruzada:

$$H(\mathbf{y}, \mathbf{p}) = -\sum_{i=1}^{n} \sum_{k=1}^{K} y_{i,k} \log(p_{i,k}) \tag{2}$$

7.1.3 One-Hot Encoding

Para representar las clases en forma numérica, se utiliza el método de codificación One-Hot:

$$C_1 = (1, 0, 0)$$

$$C_2 = (0, 1, 0)$$

$$C_3 = (0, 0, 1)$$

7.2 Redes Neuronales

Las redes neuronales son un modelo computacional inspirado en el funcionamiento de las neuronas en el cerebro. A continuación se describen las relaciones matemáticas de las redes neuronales.

7.2.1 Relación de variables

Para cada neurona, la entrada total z_j^l en la capa l es:

$$z_j^l = w_j^l a^{l-1} + b^l \tag{3}$$

7.2.2 Activación

La activación a_j^l de la neurona se obtiene aplicando una función de activación σ a z_j^l:

$$a_j^l = \sigma(z_j^l) \tag{4}$$

7.2.3 Función de costo

La función de costo C_0 mide la diferencia entre las activaciones y los valores reales:

$$C_0 = \sum_{j=0}^{n_c} (a_j^l - y_j)^2 \tag{5}$$

7.2.4 Gradiente Descendente

El propósito del algoritmo de gradiente descendente es minimizar la función de costo ajustando los pesos y sesgos. Las derivadas parciales se calculan como:

$$\frac{\partial C_0}{\partial w_j^k} = \frac{\partial z_j^l}{\partial w_j^k} \frac{\partial a_j^l}{\partial z_j^l} \frac{\partial C_0}{\partial a_j^l} \tag{6}$$

$$\frac{\partial C_0}{\partial b_j} = \frac{\partial a_j^l}{\partial z_j^l} \frac{\partial C_0}{\partial a_j^l} \tag{7}$$

$$\frac{\partial C_0}{\partial a_k^{l-1}} = \sum_{i=0}^{n_l - 1} \frac{\partial z_j^l}{\partial a_k^{l-1}} \frac{\partial a_j^l}{\partial z_j^l} \frac{\partial C_0}{\partial a_j^l} \tag{8}$$