Task 1

MNIST Image Classification Documentation

Introduction

The entire procedure for creating a high-precision image classification model for the MNIST dataset is described in this thorough documentation. This project includes data loading, careful preprocessing, model creation, efficient training methods, rigorous evaluation, and perceptive visualization.

Dataset Description

In machine learning, the MNIST dataset is a well-known benchmark dataset. It consists of 0 to 9 handwritten digits in 28x28 pixel grayscale pictures. The main goal is to group these digits into the appropriate categories correctly.

Process Overview

The project unfolds in a systematic sequence of steps:

1. Retrieved the MNIST dataset from CSV files using data retrieval.
2. Divided the dataset into input features (pixel values) and target labels (digits 0-9) using data segmentation.
3. Data normalization improves model convergence by converting pixel values to a standard range between 0 and 1.
4. Class Distribution Analysis: To understand the class balance of the data visualized the distribution of the target labels.
5. PyCaret can be used to detect anomalies by employing Principal Component Analysis (PCA) to spot potentially false data items.
6. Highlighted the data points with the highest anomaly ratings for additional examination by visualizing the most anomalous data points.
7. Principal Component Analysis (PCA) is a dimension reduction technique that can be used to reduce the dimensionality of a dataset without sacrificing important data.

8. Created a 3D scatter plot using Uniform Manifold Approximation and Projection (UMAP) to visualize the reduced-dimensional data.
9. Choosing a Sample Size: To speed up experimentation, have chosen a sample size and keep the integrity of the data.
10. Used early halting throughout the training process to reduce overfitting.
11. Thoroughly assessed the trained model's performance on the independent test dataset.
12. Calculated the key classification measures, such as recall, F1-score, and overall accuracy, such as precision.
13. Created a thorough categorization report that summarizes the model's propensity for prediction.

The Rationale for the Custom CNN Model

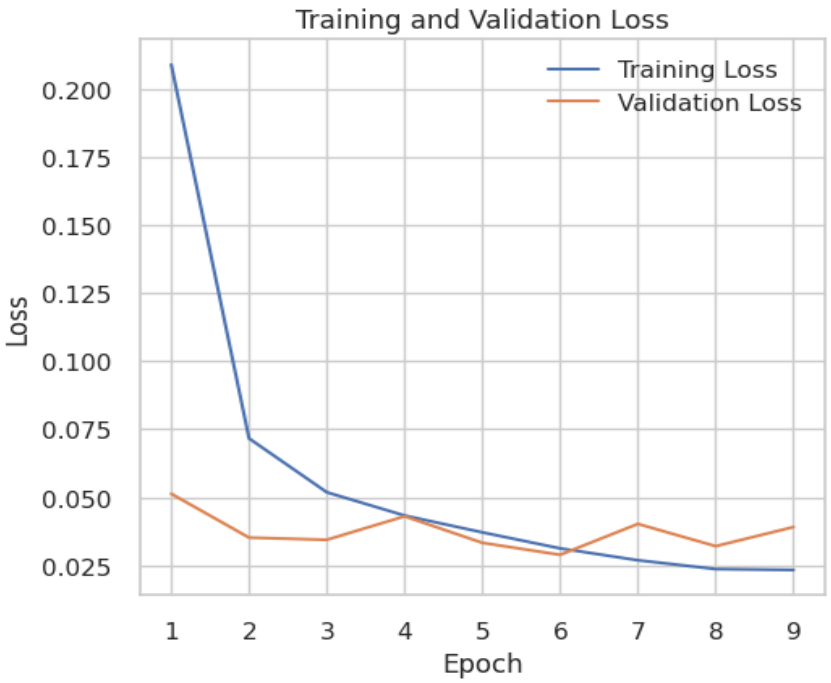
The following strong arguments support the choice to build a unique CNN model that is built especially for the MNIST dataset:

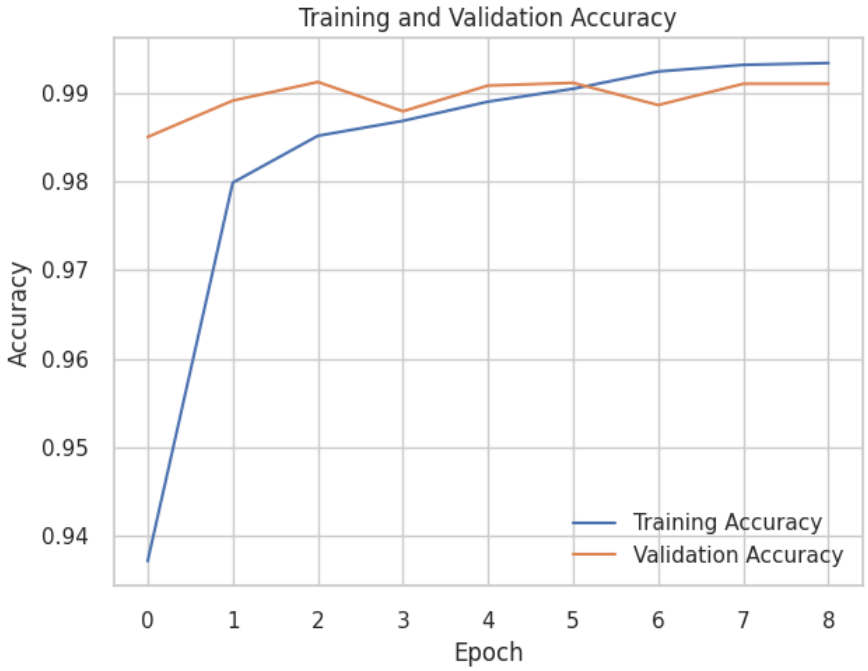
1. Custom models enable the creation of meticulously tuned CNN architectures that are specifically suited to the peculiarities of the MNIST dataset. This individualization enables the best possible adaptability to the particular job at hand.
2. Custom CNN architectures allow for a wide range of architectural configurations, hyperparameter settings, and creative architectural decisions. The investigation of cutting-edge design paradigms is made easier by this flexibility.
3. Convolutional neural networks (CNNs) and their internal workings are better understood through the development of a bespoke model. This approach improves the ability to gain insightful knowledge from model actions and results.
4. Unique dataset features or limitations can be accommodated with great ease by using custom CNN models. This flexibility guarantees improved execution on difficult assignments.
5. Adopting a unique model encourages deep learning research and invention. It encourages the investigation of novel concepts and advances knowledge in the area.

Results

Classification Report:

	precision	recall	f1-score	support
0	0.99	1.00	0.99	980
1	1.00	0.99	1.00	1135
2	0.99	0.99	0.99	1032
3	0.98	1.00	0.99	1010
4	1.00	0.99	0.99	982
5	0.98	0.99	0.98	892
6	0.99	0.98	0.99	958
7	0.99	0.99	0.99	1028
8	0.99	0.99	0.99	974
9	0.99	0.99	0.99	1009
accuracy			0.99	10000
macro avg	0.99	0.99	0.99	10000
weighted avg	0.99	0.99	0.99	10000


Training and Validation Loss
Training Loss
Validation Loss
Loss
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
1
2
3
4
5
6
7
8
9
Epoch


Training and Validation Accuracy
Accuracy
0.99
0.98
0.97
0.96
0.95
0.94
Training Accuracy
Validation Accuracy
0
1
2
3
4
5
6
7
8
Epoch

Conclusion

With this thorough documentation, the complete picture classification effort for the MNIST dataset has been methodically planned out. The project's decision to use a customized CNN model demonstrates its dedication to accuracy, adaptability, a deeper understanding of CNNs, and a desire to lead innovation in the field of deep learning.

By providing a thorough grasp of the entire workflow and the underlying justification for model selection and architectural decisions, this comprehensive documentation acts as an authoritative reference. It emphasizes the importance of striving for excellence in image classification projects.

Task 2

Database Structure Documentation

Table: USERS

The "USERS" table is a central component of the database and is designed to store information about users. It serves as a fundamental data structure for managing user records within the application. Below is a detailed breakdown of the table structure:

Fields

id (Primary Key)
Data Type: INTEGER
Description: The id field is an auto-incrementing primary key. It serves as a unique identifier for each user record in the table. This field ensures the uniqueness and integrity of each user entry.

name
Data Type: TEXT
Description: The name field stores the full name of the user. It is designed to accommodate textual data representing the user's name.

age
Data Type: INTEGER
Description: The age field is an integer data type used to store the age of the user. It captures the numerical representation of the user's age.

gender
Data Type: TEXT
Description: The gender field is a text data type that stores the gender of the user. It is intended to hold textual values, such as "Male," "Female," or other gender-related descriptors.

salary
Data Type: INTEGER
Description: The salary field is an integer data type used to record the salary or income associated with the user. It stores numerical data representing the user's salary.

Purpose

The "USERS" table is made to make it easier for the program to manage user data. It helps the program carry out routine tasks like CRUD operations—creating, retrieving, updating, and deleting user records. The way the table is set up ensures data integrity, reduces duplication, and leaves room for future improvements.

Normalization

The "USERS" table follows the rules of database normalization even though it is a simple structure. This indicates that data is set up in a way that reduces data duplication and upholds data integrity. The primary key (id) is used in this table's normalization to create distinct record identities and prevent data duplication.

Flexibility

A variety of user-related data can be accommodated by the "USERS" table's structure. As the needs of the program change, it can easily be expanded with new fields to collect additional user attributes. The table is excellent for simple user management requirements due to its flexibility and simplicity.

The "USERS" table's purpose, structure, and design principles are all well-understood thanks to its thorough description. For developers and other parties involved in the administration and upkeep of the database, it is a useful resource.

Task 3

Google Spreadsheet API Automation Documentation

Introduction

A data analysis and machine learning project called the Crop Recommendation System seeks to help farmers choose the right crops to grow based on a variety of environmental and soil factors. This documentation offers a summary of the project's features and instructions for using them. This analysis is done by automating Google Spreadsheet API.

Project Overview

Purpose

The Crop Recommendation System is made to help farmers choose crops as efficiently as possible by examining soil and environmental factors. In order to provide insights regarding crop compatibility depending on certain characteristics including soil nutrient content, temperature, humidity, and pH levels, it makes use of data analysis, data visualization, and machine learning approaches.

Key Features

- Importing data from Google Sheets
- Data visualization and analysis
- Crop recommendation models using machine learning
- Interactive graphs and displays
- Ability to change data for interaction with Google Sheets

Data Source

The project makes use of agricultural information kept in a Google Sheets file. This data is a useful tool for crop suggestion because it contains details about various crops and their characteristics.

Setup and Configuration

Prerequisites

Make sure you meet the following conditions before utilizing the Crop Recommendation System:

- You have Python 3.X installed.
- A Google Sheets spreadsheet with agricultural data is available.
- Python libraries are required, as indicated in the project code.

Library Import

Importing the appropriate Python libraries and modules should come first. These consist of tools for Google Sheets access (gspread), data manipulation (pandas, numpy), data visualization (matplotlib, seaborn, plotly), machine learning (scikit-learn), and authentication (oauth2client).

Google Sheets Authentication

Establishing OAuth2 credentials for authentication in order to access Google Sheets. In order to do this, you must provide the scope of access as well as the location of a JSON key file holding the required credentials.

Accessing Google Sheets Data

Accessing a specific Google Sheets document ("crop_recommendation") and its worksheet ("Crop_recommendation") using the gspread library. Also, printing the URL of the document and including a new worksheet ("new_analysis2").

Data Loading and Processing

Using wks.get_all_records(), the worksheet's data is loaded into a pandas DataFrame (df). This data frame includes details on various crops and the attributes that go along with them.

Data Analysis and Visualization

Utilizing several Python modules, such as seaborn, matplotlib, and plotly, to carry out exploratory data analysis (EDA) and visualization. The distribution of various properties, such as nitrogen content, phosphorous content, temperature, humidity, etc., can be seen by creating subplots and histograms. Simply selecting and printing the crops that need specific features or conditions (like low temperatures or high nitrogen content, for example), one can answer

specific queries about the dataset. Creating interactive plots that show the top crops that need particular nutrients, such as nitrogen and phosphorus.

Correlation Analysis

sns.heatmap is used to generate a heatmap that shows the relationship between the dataset's various attributes. This aids in establishing connections between qualities.

Machine Learning

Isolating the features (X) and target labels (Y) to prepare the data for machine learning. Using the scikit-learn StandardScaler to standardize. Then, using the dataset, train three distinct machine learning models (Logistic Regression, Random Forest, and Support Vector Machine) and assess each model's precision.

Model Evaluation

Creating a classification report and printing the accuracy scores for every machine learning model to evaluate each one's performance. Making a confusion matrix will help the Random Forest model see its predictions.

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	21
banana	1.00	1.00	1.00	24
blackgram	1.00	1.00	1.00	25
chickpea	1.00	1.00	1.00	20
coconut	1.00	1.00	1.00	28
coffee	1.00	1.00	1.00	16
cotton	1.00	1.00	1.00	23
grapes	1.00	1.00	1.00	21
jute	1.00	1.00	1.00	14
kidneybeans	1.00	1.00	1.00	14
lentil	1.00	1.00	1.00	16
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	18
mothbeans	1.00	1.00	1.00	16
mungbean	1.00	1.00	1.00	11
muskmelon	1.00	1.00	1.00	25
orange	1.00	1.00	1.00	20
papaya	1.00	1.00	1.00	22
pigeonpeas	1.00	1.00	1.00	13
pomegranate	1.00	1.00	1.00	23
rice	1.00	1.00	1.00	27
watermelon	1.00	1.00	1.00	23
accuracy			1.00	441
macro avg	1.00	1.00	1.00	441
weighted avg	1.00	1.00	1.00	441

Google Sheets Data Update

Searching for a specific keyword ("rice") in the spreadsheet and updating data in specific rows for "muskmelon" and "papaya."

Data Processing Approaches

Importing data from Google Sheets into a pandas DataFrame using the get_all_records function from the gspread package. This strategy works well with small to medium-sized datasets. Examining and displaying the dataset using data visualization tools like seaborn, matplotlib, and plotly. To obtain insight into the distribution and correlations of the data, histograms, bar graphs, and heatmaps are created. Using machine learning methods to complete categorization jobs. For prediction, three models—Logistic Regression, Random Forest, and Support Vector Machine—are employed. To enhance the performance of the model, data preprocessing procedures like standardization are used. Using the update_cell method from the gspread module, you may update particular data in the Google Sheets worksheet and make changes to the dataset straight from your Python code.

With the ability to update data back to the source spreadsheet, your code illustrates an organized approach to data analysis, visualization, and machine learning on agricultural data saved in Google Sheets.