Gender Classification Using Dense Neural Network

Introduction

The provided code is an implementation of a gender classification system using a dense neural network. The objective is to classify images into two categories: "male" and "female" based on their visual features.

Dependencies

The code requires the following Python libraries and modules:

os: Provides a way to interact with the operating system, used for file handling.

numpy: Used for numerical operations and data manipulation.

cv2 (**OpenCV**): Used for image processing, including image loading and resizing.

sklearn: Scikit-learn library for machine learning utilities, including data preprocessing.

tensorflow: TensorFlow, a deep learning framework, for building and training neural networks.

Dataset

The code assumes the presence of a dataset consisting of male and female images in separate directories:

male_dir: Path to the directory containing male images.

female_dir: Path to the directory containing female images.

Data Preprocessing

Images are loaded using OpenCV (cv2) and converted to grayscale.

All images are resized to a fixed size of 100x100 pixels.

Labels are assigned to the images: 1 for "male" and 0 for "female."

Data Splitting

The dataset is split into training and testing sets using train_test_split from scikit-learn. By default, 80% of the data is used for training, and 20% is used for testing.

Model Architecture

The neural network model is constructed using TensorFlow's Keras API. The model architecture consists of the following layers:

Flatten Layer: Converts the 100x100 input images into a flat vector.

Dense Layer (128 units) with ReLU activation function.

Dense Layer (64 units) with ReLU activation function.

Output Layer (1 unit) with a sigmoid activation function for binary classification.

Model Compilation

The model is compiled using the following settings:

Optimizer: Adam optimizer.

Loss Function: Binary Cross-Entropy, suitable for binary classification.

Metrics: Accuracy is used as the evaluation metric.

Model Training

The model is trained on the training dataset with the following parameters:

Epochs: 3 (You can adjust this value as needed)

Batch Size: 32

Validation Split: 20% of the training data is used for validation during training.

Model Evaluation

After training, the model is evaluated on the test dataset, and the test accuracy is computed and displayed.

Accuracy

After training, the model is evaluated on the test dataset, and the test accuracy is **86%**.

Conclusion

The provided code demonstrates the construction and training of a dense neural network for gender classification using grayscale images. The model architecture and training parameters can be customized to achieve better classification performance. Additionally, real-world applications might require further preprocessing and data augmentation techniques for improved accuracy.