Wassim Al Haraki 26/04/2023

Written Digit Classification Report

- Business Concept:

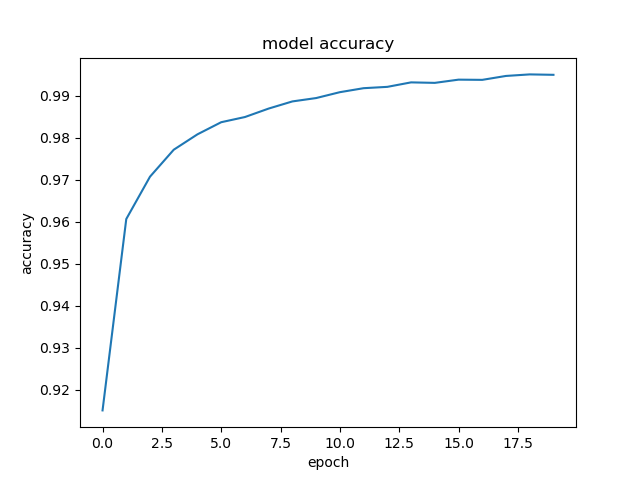
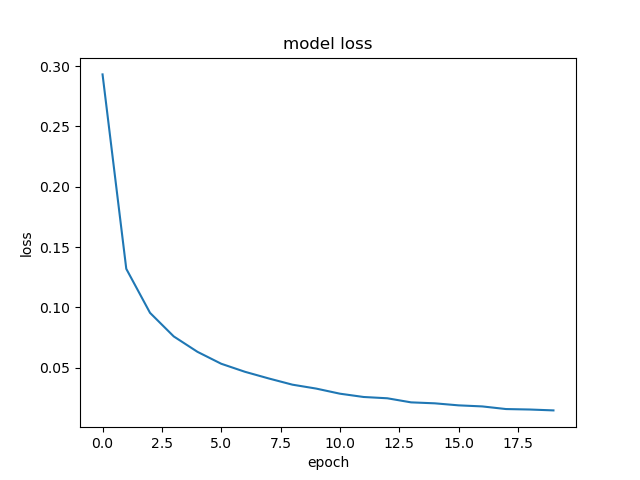
* OCR is the process of converting text in an image to a text readable/recognized by machines.
* An OCR is an essential tool for any organization that deals with large volume of handwritten documents, some of which are banks, governments, education systems, and insurance companies.
* A written digit classification model is a building block of an OCR, as converting a string to machine-readable text studies every character on its own, not the string as a whole.
* Using the MNIST dataset, a preprocessed and preformatted dataset that contains 60,000 handwritten digit images, I built and trained the model using a neural network to accurately predict digits.

- Data cleaning & preprocessing:

* The dataset I’m using is cleaned and preprocessed, but when dealing with an image classification model, there are some steps to follow:
  + In cleaning, its mainly done on the labels of the images, where any null values are dropped.
  + To preprocess images, firstly read the images using any package that supports it in python.
  + Secondly, resize the images to a common size so that the input size is constant.
  + Thirdly, greyscale the images if necessary or maintain the colors if not.
  + Fourthly, divide all elements of the image array by 255 so that the model only deals with values between 0 and 1.
  + Fifthly, the array must be reshaped to (w, h, c) where c represents the number of channels. An RGB image has 3 channels; red, green, and blue, where a greyscale image only has 1 channel.

- Model (building + training):

* Building:
  + The model I created is based on a neural network with 2 hidden layers, 1 input layer, and 1 output layer.
  + The input layer takes a list of width 28 and height 28
  + The 2 hidden layers contain 50 units each and use the ReLU activation function as its computationally light.
  + The output layers outputs an list of 10 elements, using the sigmoid activation function to output a number that represents a probability between 0 and 1, each of the 10 elements represent the probability that their respective index is the number shown in the image.
* Training:
  + The model trained for 3 minutes, running 20 epochs, averaging at 9 seconds per epoch.
  + These are some graphs that show how the model improved in the learning process.



* + The model obtained an accuracy of 99.5% on training data.

- Experimenting:

* The model scored 97.6% on testing data.
* I created a small website using flask to help understand how this model works.
* This website allows the user to draw a number on the canvas using the mouse, and on submitting, the model will predict what number he drew.
* As fun as it seems, it is not the best way to show the power of the model for several reasons:
  + Handwriting the digits and drawing them with a mouse are 2 completely different things.
  + Handwriting digits means that a pen or a pencil was used, which has different line size and gradient than the ones used in the canvas.
  + Digits in the images were centered, so if the digit drawn was not centered, result might not be accurate.