CS 417/517: Introduction to Human Computer Interaction - Project 1 (Spring 2024)
Due Date : Feb 25th, 11:59 PM

February 1, 2024

1 Introduction

In this assignment, your task is to implement a Multilayer Perceptron Neural Network and evaluate its performance in classifying handwritten digits. After completing this assignment, you are able to understand:

- How Neural Network works? How to implement Neural Network?
- How to setup a Machine Learning experiment on public data?
- How regularization, dropout plays a role in machine learning implementation?
- How to fine-tune a well-train model?

To get started with the exercise, you will need to download the supporting files and unzip its contents to the directory you want to complete this assignment.

2 Dataset

Option 1 : The MNIST dataset consists of a training set of 60000 examples and test set of 10000 examples. All digits have been size-normalized and centered in a fixed image of 28 × 28 size. In original dataset, each pixel in the image is represented by an integer between 0 and 255, where 0 is black, 255 is white and anything between represents different shade of gray. In many research papers, the official training set of 60000 examples is divided into an actual training set of 50000 examples and validation set of 10000 examples.
Option 2 : The FER-2013 dataset is a key component of emotion recognition using computer vision. Comprising 35,887 grayscale images (48x48 pixels) categorized into seven emotion classes, it offers a diverse representation of facial expressions from various demographics. Originating from the ICML 2013 Challenges in Representation Learning, this dataset is split into training, validation, and test sets for educational use.

3 Implementation

(**Notice** : You can use any library to finish this project. We recommend students to use Google Colab, which is a cloud-based service that allows you to run Jupyter Notebooks for free. To start this, follow these steps. 1. Open your web browser and go to the Google Colab website by visiting colab.research.google.com. 2. Sign up and Sign in. 3. After signing in, you can start a new notebook by clicking on File - New notebook.)

3.1 Tasks

Code Task [70 Points]: Implement Convolution Neural networks (CNN) to train and test the MNIST or FER-2013 dataset, and save the well-train model.

Code Task (1) Build your customized Convolution Neural Network (CNN)

- Define the architecture of a Convolution Neural Network (CNN) with more than 3 layers, that takes these images as input and gives as output what the handwritten digits represent for this image.

- Test your machine learning model on the testing set: After finishing the architecture of CNN models, fix your hyper-parameters(learning rate, lambda for penalty, number of layers, and number of neurons per layer), and test your model's performance on the testing set.

- Implement different optimizer (i.e., at least two). Compare the results in report and analyze the potential reasons.

- Implement different regularization methods for the Neural Networks, such as Dropout, l1 or l2. Compare the results in report and analyze the potential reasons.

Code Task (2) Fine-tune at least three different well-pretrained models (e.g., MobileNetV3, Resnet50) to get a good performance. You need to choose the specific layers to retrained and write it in the report.

Code Task (3): This code task is only for CS517. Recognize handwritten digits from a recorded video using the pre-trained model and OpenCV libraries.

Notice: The students in CS417 will get 20 points bonus if they finish this part.

```
Load the video and read frames.
Load the pre-trained model.
While the input is available, read the next frame.
    Process the frame. (options: resizing, cropping, blurring, converting to
    grayscale, binarizing, normalizing and etc.)
    Input the processed frame into the model.
    Use a threshold to detect digits.
    Put a contour around the digit and label the predicted value and probability.
    Display the frame.
Release resources.
```

Hint: Here lists some of the functions you might use.

```
cv2.VideoCapture
cv2.resize
cv2.cvtColor
cv2.threshold
cv2.putText
cv2.rectangle
cv2.imshow
cv2.waitKey
cv2.destroyAllWindows
```

Writing Report Task [30 Points]: Write a report to describe above implementation details and corresponding results.

4 Deliverables

There are two deliverables: report and code.

1. **Report (30 points)** The report should be delivered as a separate pdf file, and it is recommended for you to use the NIPS template to structure your report. You may include comments in the Jupyter Notebook, however you will need to duplicate the results in the report. The report should describe your results, experimental setup, details and comparison between the results obtained from different setting of the algorithm and dataset.

2. **Code (70 points)**

   The code for your implementation should be in Python only. The name of the Main file should be main.ipynb. Please provide necessary comments in the code and show some essential steps for your group work.