ELE547 FALL 2020 Assignment 4

Artificial Neural Network

- Visit https://data-flair.training/blogs/python-deep-learning-project-handwritten-digit-recognition/
    - This demo consists of two programs: one for training with the MNIST set, and the other is a GUI for user to input a hand-written digit.
    - These two programs require the use of "keras" which is a high-level API of tensorflow. Depends on your Python IDE, you may need to install "tensorflow" and "keras". There may be version compatibility problems between tensorflow and keras. The solution is either to upgrade tensorflow or to downgrade keras, whichever will work for your environment.
    - The training on MNIST set (60,000 patterns) for 10 epochs will need some serious computational power and need to run for long time. Fortunately, this is a "one-time task" and therefore should be run on a PC (Linux, Windows or MacOS).
- Run the training program on a PC to obtain the trained model: mnist.h5.
- Modify the training program so the activation function of the neural network changes from 'relu' to 'sigmoid'. Save the trained model to: mnist.sig.
- Run the GUI program to test the two versions of neural networks (relu vs. sigmoid). Write down your "user impressions" in the lab report.
- Try running GUI program on Raspberry Pi (without sense hat) and report the user experience in the lab report. Can you sense any difference?

For Lab Report/Presentation:

1. Written report for this lab:
    a. A short essay on the user experiences of the GUI program in four different settings: relu vs. sigoid, and PC vs. Raspberry Pi.
    b. The python program of your final GUI on Raspberry Pi.
2. Demonstrate/show the running of the best user GUI version on Raspberry Pi.

Bugs, Fixes and Platform considerations

1. My Working environment
    a. Windows 10
    b. Anaconda → Spyder 4 → Python 3
2. Tensorflow and Keras
    a. Tensorflow has already been installed with Anaconda installation.
    b. Install keras via "pip". When running the program got a warning about tensorflow must be version 2.2 or higher.
    c. Downgraded keras to a lower version and solved this problem.
3. Initial bugs
    a. Variable "num_classes" was used before declared. Move the declarations up and solve this little bug.
4. Window DPI
    a. After the neural network is trained, run the GUI program and the results are obviously wrong. Use "img.show()" to see what was grab from screen via "im = ImageGrab.grab(rect)" and found the screenshot is off.
    b. Suspect that this is due to the wrong coordinates provided by the "rect = win32gui.GetWindowRect(HWND)". Found the solution here: https://stackoverflow.com/questions/40869982/dpi-scaling-level-affecting-win32gui-getwindowrect-in-python
5. Reversed polarity
    a. The result is still not very good. From the screenshots, I realized that the GUI is using background white, while the training set is background black. In other words, the GUI uses a negative image compared to the MNIST training set. Change the "img = img/255.0" (scale/normalize to 0 and 1 range) to "img = 1-(img/255.0)".
6. GUI program Windows dependency
    a. Finding Linux equivalent is the key to run GUI on Raspberry Pi. What is needed here is to find the "window" ID in HWND and then use that to get the coordinates of the window as "rect". Possible alternative: (no need to involve win32gui)

```
box =
(self.canvas.winfo_rootx(),self.canvas.winfo_rooty(),self.canvas.winfo_rootx()+self.canvas.winfo_width()
,self.canvas.winfo_rooty() + self.canvas.winfo_height())
im = ImageGrab.grab(bbox = box)
```

    b. "ImageGrab" from PIL (pillow) is not available for Linux! May use "pyscreenshot" https://github.com/ponty/pyscreenshot
    c. Missing h5py:
        i. Sudo apt-get install libhdf5_dev
        ii. Pip install h5py