Angel Candelas

Prof. P. McManus

ITAI 1378

June 21, 2024

L05 Image Classification with SVM

I’ll preface this journal entry by stating that I’m starting to feel like I skipped an intro class that I did not know about or was not offered during the summer semester before enrolling in this class. The reason I say that is because this lab proved to be a lot more difficult than I was expecting. Watching the recorded class session helped with navigation a lot, but I felt lost for most of the lab. With that said, I’ll dive into what I did learn going through this experience.

This was the first time I learned about the Support Vector Machine algorithm and the CIFAR-10 dataset. From my understanding, SVMs are a very good tool that can be used to classify different things. They can be used in a variety of different ways including object and facial recognition, medical imaging, and recognizing handwriting. Using the CI-FAR 10 Dataset file, I went through the different steps involved in using an SMV. The first step involved installing and importing the necessary libraries that were needed. For this lab, there were a total of 4 libraries; numpy, matplotlib, tensorflow and scikit-learn. I knew about pip installing from the previous lab, but what something new that I learned with this lab is that we can install multiple libraries at once by using one line of code and separating the names of the libraries with a space in between. The next step was to load and preprocess data. In this step we are, first, importing the data and then preprocessing it to scale it and prevent larger files from affecting performance. Finally, you’ll split the data into training and testing sets. I did not quite understand this (and I’m not sure that I do even now), but I looked it up a few different times and this was my understanding of it after reading about it. Step three was training the SVM model to recognize patterns in the data. You do this by choosing a kernel which changes the input date to help find separation for things that may not be linearly separable. The kernel will calculate similarities in the data points and use them to find the best hyperplane to separate them.

While following along with the recorded class session, I was able to successfully install and import the libraries. I was also able to click through the different codes and see the results (pictures #1 and #2 of the results included at the end of this journal entry), but what I was not able to figure out was how to get different results. I’m not sure if the intention was to only get the same results, or to try and change the code to get something different (I seemed to break it anytime I changed anything).

My biggest takeaway from this lab is that there is so much out there that I have yet to learn, and this gave me a good starting point. Reading through the powerpoints and then looking things up for more information are my next steps so that I can hopefully get through the next lab with less difficulty.

Picture #1: Picture #2:

A screenshot of a computer screen

Description automatically generatedA screenshot of a test

Description automatically generated