Kaire Campomanes

Professor Patricia Mcmanus

ITAI 1378

26 September 2024

Reflective Journal

This lab project aimed to teach me how to understand and use a Support Vector Machine (SVM) for photo categorization using the CIFAR-10 dataset. CIFAR-10 is a set of tiny images divided into ten divisions, such as cars, pets, and aircraft. I worked with a subset of three categories, ships, dogs, and cats for this lab. Using a Support Vector Machine model and measuring its classification accuracy with these images was the goal. After completing this assignment, I was able to recall the fundamental points behind collecting data, training an SVM model, and evaluating the model's effectiveness.

First Impressions

I was excited to learn, but also scared when I initially saw that I had to develop a Support Vector Machine (SVM) for image classification using the CIFAR-10 dataset. I don't have any previous tech background or any in-depth knowledge of coding. Although I hadn't used SVM previously, I do keep up with updates periodically about machine learning and how it can be used to help with classification. Classifying photographs of ships, dogs, and cats using this technique seems like a difficult task. I didn't know much about SVM, despite having some general knowledge of picture categorization from earlier research in class. I was mostly nervous because of my little experience and I realized I would have to start from scratch.I was eager to get started so I could understand how machine learning models can be applied in different settings.

Data/Model Training & Testing

The optimal hyperplane to arrange many data classes is found using the algorithm. SVM is used to make use of the gaps in various data sets to improve prediction accuracy. In order to sort through the grayscale photos of ships, dogs, and cats that I downloaded from the CIFAR-10 collection, I had to learn how to use an SVM. SVM often performs well in situations where data can be divided linearly. For this challenge, I applied the simplest sort of kernel, the linear kernel. The model's accuracy was around 55-60%, which is not very high for image classification but is expected given the simplicity of the SVM and the dataset. The model worked well in differentiating between ships and other classes but had some trouble with cats and dogs, which are more visually similar. SVMs can be great for some tasks but may not be the best for image classification compared to more visually specific algorithms.

Challenges Faced

One of the challenges I faced was the amount of time it took to figure out the SVM. I have no background with coding, and it took me a few tries to get it right. I even looked at the class GroupMe conversation to help with some confusion I had. I became overwhelmed and had to take a few breaks in between attempts. I had a difficult time when I was trying to understand the computational demands of how to train the SVM on a large dataset like CIFAR-10. Even after simplifying the problem by converting the images to grayscale and reducing the dataset to just three classes, it still took more time than I anticipated. In the initial stages, I forgot to flatten the grayscale images, which caused an error when trying to fit the SVM model. Experiencing this mistake made the importance of checking the format of data before applying algorithms clear. I also encountered some unexpected results. I was shocked with how low the accuracy was when predicting between cats and dogs. This taught me that some classes are just harder to separate than others. One of the most rewarding parts of the lab was seeing the SVM model make predictions for some images. It was satisfying to see the results of the model's training, especially after going through the data preparation and training process.

Learning Process

Working through the provided Jupyter notebook was enjoyable. The CIFAR-10 dataset was first easy to load and view. I knew what it meant to convert images to grayscale and then normalize the pixel values. Additionally, as SVM frequently requires numerical data in a specific format when working with picture datasets, it was logical to flatten the photos into single-dimensional arrays. However, some aspects of the task, such as understanding SVM's core operations, proved more challenging. Finding the best hyperplane to divide data points into higher dimensions first appeared a bit vague.

I had a few "aha!" moments while going through this assignment. One big realization was understanding the operation of the SVM kernel function. The reason we needed to use multiple kernels became clear to me when I ran the code using the linear kernel and saw the results. Kernels allow SVM to handle a variety of data separation scenarios. Even though I stuck with the linear kernel due to its convenience of use, this made me want to try other ones.

In order to understand the mathematics underlying SVM better, I looked up additional information, particularly about the optimization of the margin between classes. I felt more at ease after doing this study because the supplementary reading clarified a few points that the notebook didn't fully cover.

Responses to Lab Questions

(I did not see any specific lab questions in the Jupyter Notebooks, so I came up with these two on my own.)

What kernel did you choose, and why? Since the linear kernel is straightforward and effective in situations when the data is linearly separable, I went with it. In addition, the linear kernel uses less computing power than more sophisticated kernels like RBF, which, given my lack of experience with this, I thought was more reasonable.

What are the strengths and weaknesses of SVM in image classification?

SVMs fail miserably at image classification tasks where the associations between pixels are important, such as differentiating between dogs and cats. SVMs use a lot of computer power when used on large datasets as well. They would be better with smaller dataset tasks and with less complexity.

Personal Growth

This exercise has impacted my understanding of image classification and machine learning. I now have a better grasp of how SVM works and how it can be applied to classification tasks, even though it may not always be the best choice for image data. I learned that while SVM is a solid algorithm for certain problems, it has limitations when it comes to tasks like image classification, where some spatial relationships between pixels matter.

I’ve gained new skills in working with image data, particularly in how to preprocess images for machine learning models. I now feel more confident in my ability to apply SVM to other classification tasks, although I recognize that I would need to carefully consider the type of data I’m working with. I also learned about the importance of choosing the right model for a given task—SVM is good for some applications, but I’m now more interested in exploring more advanced algorithms like Convolutional Neural Networks (CNNs).

Looking Ahead

Once I've completed this task, I still have some questions. For example, I'm curious about how well SVM might work with different kinds of kernel functions or with more advanced techniques for feature extraction. Furthermore, I'd like to know how SVM compares against CNNs on progressively complex datasets.

In the future, I'm looking forward to learning advanced machine learning techniques, particularly ones that involve computer vision and image processing. CNNs are ranked highest in my rating because they seem to be more appropriate for image classification jobs. I'd also like to know more about how different algorithms handle larger datasets and increasingly complex images.

Works Cited

JupyterLab Documentation

"Starting JupyterLab." *JupyterLab Documentation*, Project Jupyter, https://jupyterlab.readthedocs.io/en/stable/getting_started/starting.html. Accessed 26 Sept. 2024.

GeeksforGeeks Article

"Classifying Data Using Support Vector Machines (SVMs) in Python." *GeeksforGeeks*, 6 Apr. 2023, https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-python/. Accessed 26 Sept. 2024.

PyTorch CIFAR-10 Tutorial

"Training a Classifier." *PyTorch Tutorials*, PyTorch, https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html. Accessed 26 Sept. 2024.

(I also watched the supplemental resources for each module.)