Misty Richardson
ITAI 1378

L04 Image Classification with k-Nearest Neighbors

In this journal, I will describe my learning path of a lab experience on the k-Nearest Neighbors algorithm (k-NN) implementation for image classification. The lab used the MNIST dataset, which is a collection of handwritten digits that help in comparing various machine learning algorithms. In this task, it was not only about implementing the model, but in depth understanding of how it works, the preparation needed for the data, and critically evaluating the model's performance.

My understanding of the KNN algorithm is that it will take a picture of whatever you uploaded and will try to match it with other images to figure out what the image is.K-Nearest Neighbors algorithm is a simple yet effective algorithm that is an integral part of the machine learning toolkit. There's a classification, which votes for a new example by the decision of the 'k' closest neighbors out of the training set. What is compelling about k-NN is its simplicity and efficiency, particularly in image classification where the patterns are not linearly separable. The MNIST example is an image that can be classified depending on the images that closely resemble the known examples.

First, the lab was to load MNIST dataset, a category that gave me a glimpse of what image data looks like and how data structure is built. One of the most valuable steps in this phase was visualizing the images because it was the point where I actually felt the data and what it would be like training my model to classify it. Training and testing partitioning got ready for our model and the evaluation could be independent from the training set to remove the bias. This phase asserted the necessity of data uniqueness for developing the model and to ascertain the level of generalization of the model precisely.

The selection of the right 'k' value turned out to be the most pressing problem. At first, my model had low values for accuracy scores. Through experimenting with multiple k values and watching its influence on the performance, I knew for sure that is essential to the process of machine learning. The other problem that I faced was why some images were wrongly classified. But in reality when I inspected all of them helped me to perceive limitations of k-NN, except when handwriting was very clear or when data had unusual representation.

Responses to Lab Questions:

Misty Richardson
ITAI 1378

Q1) If you were dealing with a much larger dataset with more features what strategies could you use to reduce the computation time of the KNN algorith?

A1) In the training process I would upload images that were unquestionable of what it was. What I mean is I would upload only images of a cat or dog and then after the initial training was done I would then start to upload images of other cats or dogs but with "distractions" around it like trees, toys, or humans. I then would start the training again with the new images. I believe that if the KNN algorithm can be aware of the distractions and look past that then it would be able to compute faster.

Q2) Research the concept of overfitting the underfitting. How might these issues manifest in a KNN model, and what strategies can be used to address them?

A2) Overfitting imply that the model is well on the training data but has poor performance when new data is coming. Underfitting refers to a model that is not good on the training data and also cannot be generalized to predict new data.

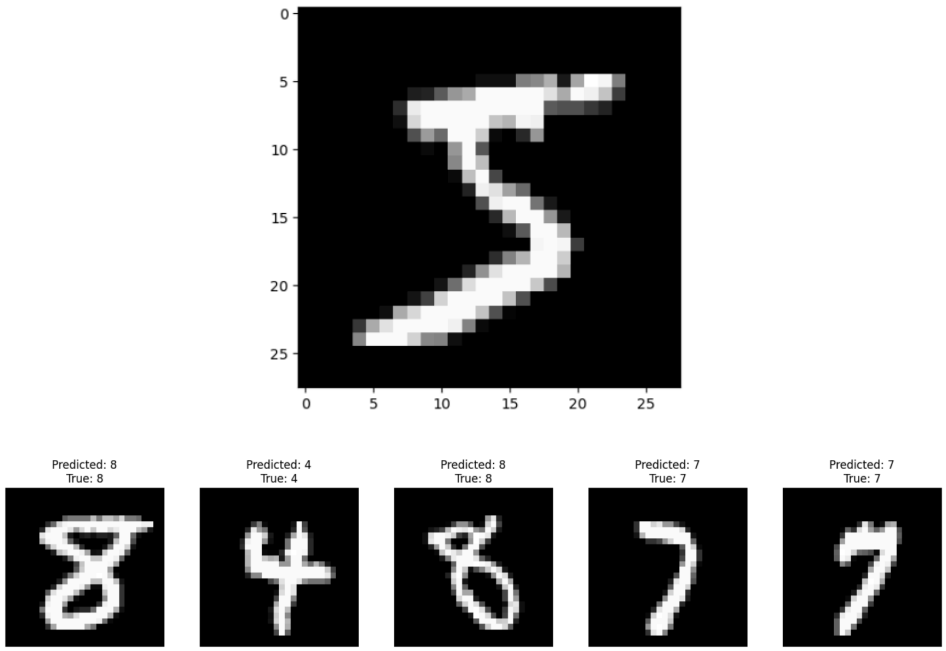
Q3) Discuss the pros and cons of the KNN algorith based on your experience in the lab. Consider aspects such as a computation time, scalability, and accuracy

A3) The few times I used the KNN algorithm I noticed that it was not registering the "distractions" or was TOO distracted by the "distractions that it did not compute correctly.

Q4) The Knn algorith is considered a lazy learning algorithm. What does this mean? And what are the implications of this characteristic in terms of model training and prediction?

A4) KNN (nearest neighbor) is known as a lazy learner. Here, the algorithm does not generalize. On the contrary, it only sees the training points used without performing generalizations. This requires different types of learning as different languages have different structures, and there is no direct training phase. There is no generalization because in KNN we keep all training data. It is a non-parametric learning algorithm as it has no assumptions about the data distribution.

Misty Richardson
ITAI 1378



References

- 43-Image Classification with KNN| Comprehensive Guide| K-Nearest Neighbors (KNN)…
- K Nearest Neighbors | Intuitive explained | Machine Learning Basics
- What's the KNN?. Understanding the Lazy Learner… | by Jisha Obukwelu | Nerd For Tech | Medium