FINAL PROJECT

STA 141A

Due Monday, June 11 by 5:00 pm

Description

The dataset consists of 60000 32×32 color (RBG) images in 10 classes, with 6000 images per class. The 10 classes are: **airplane, automobile, bird ,cat, deer, dog, frog, horse, ship and truck**. There are 50000 training images and 10000 test images. The main purpose of this final project is to develop a machine learning algorithm that can automatically recognize test images. The machine learning algorithm is trained by the large amount of training images and we explore different ways to optimize the algorithm.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

The classes are completely mutually exclusive. There is no overlap between automobiles and trucks. "Automobile" includes sedans, SUVs, things of that sort. "Truck" includes only big trucks. Neither includes pickup trucks.

Your data source is the binary version that contains the files `data_batch_1.bin`, `data_batch_2.bin`, ..., `data_batch_5.bin`, as well as `test_batch.bin`. Each of these files is formatted as follows:

$<1\times$ label$><3072\times$pixel$>$

...

$<1\times$ label$><3072\times$pixel$>$

In other words, the first byte is the label of the first image, which is a number in the range $0-9$. The next 3072 bytes are the values of the pixels of the image. Notice that each image is a 32×32 RBG (Red, Blue and Green) color image. The first 1024 bytes are the red channel values, the next 1024 the green, and the final 1024 the blue. The values are stored in row-major order, so the first 32 bytes are the red channel values of the first row of the image.

Each file contains 10000 such 3073-byte "rows" of images, although there is nothing delimiting the rows. Therefore each file should be exactly 30730000 bytes long.

There is another file, called **batches.meta.txt**. This is an ASCII file that maps numeric labels in the range $0-9$ to meaningful class names. It is merely a list of the 10 class names, one per row. The class name on row i corresponds to numeric label i.

Questions

Use R to find answers to all of the following questions (that is, don't do any by hand or by point-and-click). Save your code in an R script. Try to complete at least one every day until the assignment is due. Some problems are more difficult than others, while some are lengthier than others. Some problems may be easier if you have completed earlier ones. It is recommended that you read all the questions before beginning the problem set so you are able to adequately pace yourself.

You may NOT use packages for k-nearest neighbors and cross-validation in this assignment. All other packages (for example, ggplot2) are okay.

1. Write a function `load_training_images()` that loads the training images and the corresponding labels from the 5 provided training binary files (not the testing bin file), binds them into one data type (a list or a data frame or a matrix) and saves this data type to an RDS file. Your function should have arguments to set the path for the input directory and the output RDS file. Keep your function short and simple by using an apply function or a for loop rather than repeating code.

   Write a second function `load_testing_images()` that loads testing images and labels, binds them into one data type (list/data frame/matrix) and saves this data type to another RDS file. *No interpretation is necessary for this question.*

2. Write a function `view_images()` that displays one observation (one image) from the data set as a color image and its corresponding label (ex: airplane, bird,cat...). Your function should allow users to specify which observation they want to display. *No interpretation is necessary for this question.*

3. Explore the image data. In addition to your own explorations:

   Display graphically what each class (airplane, bird,cat...) looks like. You can randomly choose one image per class.

   Which pixels at which color channels seem the most likely to be useful for classification? Which pixels at which color channels seem the least likely to be useful for classification? Why?

4. Write a function `predict_knn()` that uses k-nearest neighbors to predict the label for a point or collection of points. At the least, your function should take the prediction point(s), the training points, a distance metric, and k as input. *No interpretation is necessary for this question.*

5. Write a function `cv_error_knn()` that uses 10-fold cross-validation to estimate the error rate for k-nearest neighbors. Briefly discuss the strategies you used to make your function run efficiently.

6. Display 10-fold CV error rates for $k = 1, ..., 15$ and at least 2 different distance metrics in one plot. Discuss your results. Which combination of k and distance metric is the best? Would it be useful to consider additional values of k?

7. For each of the 3 best k and distance metric combinations, use 10-fold cross-validation to estimate the confusion matrix. Discuss your results. Does this change which combination you would choose as the best?

8. For the best k and distance metric combination, explore the training data that were misclassified during cross-validation by displaying a confusion matrix. Discuss what you can conclude about the classifier.

9. Display test set error rates for $k = 1, ..., 15$ and at least 2 different distance metrics in one plot. Compare your results to the 10-fold CV error rates.

10. Briefly summarize what each group member contributed to the group.

    Assemble your answers into a report. Please do not include any raw R output. Instead, present your results as neatly formatted[1] tables or graphics, and write something about each one. You must **cite your sources**. Your report should be **no more than 12 pages** including graphics, but excluding code and citations.

    ### What To Submit

    Email a digital copy to `spring18stat141a@gmail.com`. The digital copy must contain your report (as a PDF) and your code (as one or more R scripts).

[1] See the graphics checklist on Canvas.

Additionally, submit a printed copy to the box in the statistics department office[4]. The printed copy must contain your report and your code (in an appendix). Please print double-sided to save trees. It is your responsibility to make sure the graphics are legible in the printed copy!

Hints

- The `grid.raster()` function displays a RBG color image.
- There's also a built-in function for computing distances.
- It's a good idea to break the steps in `predict_knn()` and `cv_error_knn()` into even smaller functions that those functions use.
- Computing distances is time-consuming, so avoid doing so in a loop.
- Ties in k-nearest neighbors can be broken by random selection, by choosing the most popular class, or by other strategies. Some strategies are more effective than others.
- The `rep_len()` function is useful for assigning a cross-validation fold to each observation.
- Rather than splitting entire observations into folds for cross-validation, it is easier and more efficient to split their indexes (row numbers) into folds.
- A confusion matrix shows the frequencies (or proportions) of predicted class labels versus true class labels. A confusion matrix provides more information about a classifier's strengths and weaknesses than the error rate alone.
- A heatmap is useful to display a confusion matrix when the matrix size is relatively big.

[4] 4th floor of Mathematical Sciences Building