CIFAR-10 Tiny Image Classification

Michael Hemphill

ME 532 Section 004

mhemphill2@wisc.edu

https://github.com/mhemphill2/ME532FinalProject

Data set

https://www.cs.toronto.edu/~kriz/cifar.html

I plan to use the CIFAR-10 tiny images data set for the final project. The dataset contains pictures with 1024 pixels (32x32) and has a total of 60,000 pictures involved. The data is split so that 10,000 images are for testing and the others are for training. If necessary, these numbers can be changed when implementing the program. These 60,000 images are evenly divided into 10 separate classes. The 10 labels are airplane, automobile, bird, car, deer, dog, frog, horse, ship, and truck.

The data set is a difficult one to reach near 100% classification rate on due to the low resolution of the images. The primary goal is to maximize classification rate amongst the test set. An additional option may be introduced so that after the algorithms are trained, it will be possible to input further user provided images and test those against the classifiers.

Algorithms

First the data will require some initial preprocessing. This will likely include normalization of the data for pixel image values from 0 to 1. The images may be reduced in size depending on the size of the main feature we are trying to classify compared to all the background. Additional steps will likely be taken but because the images are already very small in size, not much preprocessing is likely to be necessary.

The first algorithm to be implemented will be ridge regression with regularization. Cross validation will likely be used with this to find an optimal lambda and bias-variance tradeoff. The second algorithm will be the K nearest neighbor classification. This will still be supervised learning as opposed to K-means clustering. The last algorithm to be used will be neural networks via backpropagation. It will likely have from 2 to 10 hidden layers and the number of neurons per layer is undetermined at this time. Further properties are to be determined.

Because the dataset is well known, there is likely to be testing results and error data available for machine learning algorithms used by others. If this is the case, validation will be done by obtaining various classification results found online and comparing the performance of my algorithms to others. This should inform me if the results I obtain are relatively accurate or farfetched for the data used and classification rates obtained. Beyond this external review, I will examine and compare the classification rates amongst my own 3 algorithms to determine which is most effective. This process will involve changes of variables within the algorithms in order to most efficiently optimize the learners for the task at hand. All algorithms will be written in Python.

Timeline

- Data input, Preprocessing – 1 week (accomplish by ~ Oct 30)
- Implement K-NN algorithm – 1 week (accomplish by ~ Nov 6)
- Implement ridge regression algorithm – 1 week (accomplish by ~ Nov 13)
- Implement neural network – 1 week (accomplish by ~ Nov 20)
- Validation, comparison of results, tweaking algorithms – 1-2 weeks (accomplish by ~ Nov 27)
- Final report – 1-2 weeks (accomplish by ~ Dec 11)

The timeline is a rough estimate of the necessary tasks to accomplish and the planned dates for the completed tasks. It is subject to change and I believe that attempting to accomplish everything related to coding by the end of November gives some flexibility for milestone deadline fluctuations during the development of the program. After this, 2 weeks should be adequate for the final project report. None of these are hard set deadlines as the code development will likely involve overlapping some of these tasks. The Github link is provided on the cover page and will be updated throughout the development process.