

## Domain and Dataset

Domain: This project's area of interest is image recognition using convolutional neural networks (aka ConvNets or CNNs).

Dataset: CIFAR-10 (<https://www.cs.toronto.edu/~kriz/cifar.html>) is a dataset of 60000 images with a size of 32x32 pixels, spread across 10 different categories (e.g. airplane, bird, dog, etc.).

## Problem Statement

Problem Statement: Solve an image classification task, the CIFAR-10 dataset, using convolutional neural networks.

Specific Goals:

- 1) Get a "working" understanding of both neural networks and convolutional neural networks.
- 2) Obtain a respectable accuracy score on CIFAR-10 (see Benchmark section below for my definition of "respectable") with a ConvNet of my own design.

## Solution Statement

The main package that I used was Keras, using the TensorFlow backend. Also, I wanted to run this on a GPU (graphics processing unit), so with the (absolutely critical) help of Joshua Cook, I managed to setup a Nvidia-Docker image that ran on AWS (Amazon Web Services).

## Exploratory Data Analysis (EDA)

Because the CIFAR-10 is a dataset that comes prepackaged with Keras, there may not have been a need for any significant EDA on the images. However, I did wish to know the structure of the data (e.g. how each image is represented by arrays) and get some sense of what the computer was seeing.

Therefore, for a discussion on the EDA that I performed on this dataset, please see the section "Mine: Perform & summarize EDA" in the notebook entitled "Part 2 - Image Classification Capstone - John Hou". This notebook is located in the part-02 subdirectory of the `general_assembly_rubrics_for_individual_parts` parent directory.

## Metric

For a metric, the obvious one is accuracy: the ability of the ConvNet to correctly categorize the different images into the correct classes.

Of course, since the other goal in my project was to develop a good understanding of CNNs, there is another metric that I have to consider: how well have I understood neural networks, and am I able to explain them in something that resembles Earth English?

This second metric is just as important to me, if not more so, than the first. It goes without saying that this metric of understanding the material will be much harder to measure.

## Benchmark

Rodrigo Benenson's github: ([http://rodrigob.github.io/are\\_we\\_there\\_yet/build/classification\\_datasets\\_results](http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results)) is an excellent resource that lists the results, and their respective white papers, of other people's efforts at various datasets. Of course, the CIFAR-10 is among one of these datasets, and the lowest score posted there is 75.86%. Given the short(er) time frame to learn about neural networks (mostly on my own, but not without some help from Joshua Cook and Mike Frantz), my benchmark was to at least match this lowest score.

## Results - Discussion

Joshua Cook suggested that before I make the deep dive into neural networks, I should use a simple sklearn logistic regression on the dataset. Though this was an interesting exercise, the logistic regression method scored an unimpressive 32% test accuracy (again, please see the "Part 2 - Image Classification Capstone - John Hou" notebook that I referenced in the EDA section above).

Moving on to the actual ConvNet, I obtained a validation accuracy of approximately 61%. Though this wasn't quite enough to match our benchmark of 75.86%, I am not completely dissatisfied with the results for several reasons:

- 1) The convolutional neural net architecture that I implemented was fairly standard and not overly complex. Two full cycles (where each cycle consists of 2 convolutions, each followed by a standard ReLU activation, which was then followed by a MaxPooling and Dropout) with a fully connected layer at the end is very basic. Please see either the `cifar10fromkeras.py` in the **code** folder or the `screenshot_convnet_arch` image in the **misc** folder to see the structure.
- 2) Due to consistent problems (see Challenges section below), I was only able to successfully run 20 epochs. Even with this limited approach, my validation accuracy started at 22% for the 1st epoch and eventually reached 61% by the 20th epoch. Clearly, the ConvNet was learning, and it is clearly possible that it could've have learned more, given more epochs.
- 3) I was not able to experiment with parameter tweaking, which may have made a difference in the results.

## Challenges

I ran into several issues along the way:

- 1) Keras would **not** run in a Jupyter notebook without crashing. Again, thanks to Joshua Cook, I wrote a script, containing my Keras code, that would run through docker bash.
- 2) Even as a script, if my neural net had anything more than 24 filters at the first convolutional layer, it would **not** run for more than 10 epochs. I finally had to reduce the number of filters to 8.
- 3) I spent much more time than originally planned putting a website (using Flask) together for my presentation.
- 4) I really made an effort to understand the nitty gritty details under the hood regarding neural networks and convnets specifically. This was quite a challenge. Even though I utilized the well constructed Stanford CS231N course (<http://cs231n.github.io/>), I believe that this course may have been too much to digest in a short time period. This course also isn't the best course for beginners like myself. I spent too much time trying to understand dense material, instead of spending that time tinkering with Keras (which would have been a wiser option, given the limited time frame).

## Conclusion and Next Steps

Though it wasn't easy to learn difficult material in a short time span, I thoroughly enjoyed exploring the world of convolutional neural networks and image recognition. I want to continue learning about what exactly is going on under the hood of neural nets, how to build more effective neural nets, and how to tweak neural net parameters without blindly shooting into the abyss.

Another area that I'd like to focus on is TensorFlow. Keras is great, and I plan on using it since it apparently makes life so much easier, but understanding TensorFlow (and utilizing it effectively) is one of my primary goals for the future.

Lastly, I want to say that a mere 3 weeks ago, I had no idea what to focus on for my capstone project. Nothing grabbed my attention. Fortunately, Joshua Cook recommended Neural Networks, Convolutional Neural Networks, and Image Recognition, which is a fascinating area of study (and fairly new). He was also the one that framed my capstone as an opportunity to really explore something on my own. I thank him for introducing me to this subject and for helping me along the way.