Multiclass Image Classification using Neural Networks (ME397 Final Report)

Akshay Kumar Varanasi

17th December 2018

1 Introduction

The problem statement which I was interested is Multi-class classification using Neural Networks on large datasets. Knowing that deep learning and Neural Networks are of broad and current interests, I wanted to learn it through this project. I also wanted to learn PostgreSql in the process and tried to do my best to use it.

2 Dataset

Dataset that I wanted to use was Kaggle-Inclusive Image Challenge. This is Kaggle challenge by Google AI to develop models that are robust to blind spots that might exist in a data set, and to create image recognition systems that can perform well on test images drawn from different geographic distributions than the ones they were trained on. For testing on this data, I had to train on Open Image Validation dataset(12GB). The problem which I faced due to large dataset is I could train only on few images and so instead of testing on Kaggle dataset, I tested on other part of the same dataset from which I am training. One reason for doing it was to see how my model is performing as I have the labels but I don't have for Kaggle test data. Another obvious reason was the size of the Kaggle test dataset.

3 Creation of Database Table

Since I am working with real world data images which is really really large , I thought of using creating database for the dataset. Since this is image dataset, doing it is bad idea due to the large amount of space they take up, which can affect database read performance ,making it take longer and cost more to store. Instead, it's best to store the images elsewhere and then store a reference to them in your database. So, I accessed the images directly from the folder but since I said I will use database for reading I used it for accessing the database for its labels. In the folder, the code for creating the table is also included. The labels data for it is 'validation-annotations-human-imagelabels.csv'. For doing this, I had to install "RPostgreSQL" package and use library "DBI" for connecting to database. After connecting to database, I read the required table and convert it into dataframe to do further processing. (Look at the Rmd file)

4 Training Data

For training, I pick random 20 classes, which are given below with descriptions from class-descriptions.csv,out of around 1000 plus classes. I bring down the number of images to around 60,000 from around 500,000. But still 60,000 is big enough for my laptop to handle, so I chose to train on 5,000 images and use 2000 for validation.

	Class	Description
1	/m/01g317	Person
2	/m/09j2d	Clothing
3	/m/04yx4	Not available
4	/m/0dzct	Human face
5	/m/07j7r	Tree
6	/m/05s2s	Plant
7	/m/03bt1vf	Not available
8	/m/07yv9	Vehicle
9	/m/0cgh4	Building
10	/m/01prls	Land vehicle
11	/m/09j5n	Person
12	/m/0jbk	Clothing
13	/m/0k4j	Not available
14	/m/05r655	Human face
15	/m/02wbm	Tree
16	/m/0c9ph5	Plant
17	/m/083wq	Wheel
18	/m/0c_jw	Furniture
19	/m/03jm5	House
20	/m/0d4v4	Window

Table 1: Class descriptions

5 Building Neural Network Model

Here comes the main part which is to build neural network for doing the classification. Since this is Image Classification Problem, it is better to use CNN as it has been shown that they are good for those tasks. I build the following model with some intuition and some experiments with the parameters like filters, units, dropout etc. I read about these things in the links provided at the end. I used 5 Convolution layers to extract the features and 7 dense layers in the model. In between we need to flatten the features to give it as input to dense layers. We use dropout to prevent over fitting and use ReLU for sparsity. Parametric ReLU was used due to its advantages such as faster training and fixing dying ReLU problem.(See ReLU link)

After building the Neural Network, I compile it with appropriate loss function, optimizer and the metric. In this case, I used categorical cross entropy, since cross entropy is preferred loss function for images. And since this is classification problem we use categorical one. Metric which we are interested is accuracy so we use that and optimizer used was Adam. After reading about various optimizations in overview of the gradient descent algorithms(see the link), I felt that is the best.

I fit the model. I set the batch size as 300 and run 20 epochs. I tried different sizes for batches and different epochs. The More, the better but due to computational limitations could not go further.

6 Testing

From the 60,000 images of 20 classes, I picked 2000 images for testing. I read the testing data images and make it into an array. I input it to neural network model and get the prediction as probability of an image belonging to each class. I give the image a class which has the highest probability. In this way, I do Multi class image classification.

7 Results

After I get the predicted labels, I compare it with available labels to get accuracy. In my case evaluated accuracy is of 0.173 and calculated accuracy is of 2.7% . It is quite low due to various reasons such as.

- Not enough Training data given the complexity of the problem and how the data looks.
- One image can have multiple labels like the figure 1 can be person and human face. Since this data is labeled by different humans, the labels are not dependable.
- Maybe my model is not enough good. With my limited knowledge, I could only build this mediocre model.

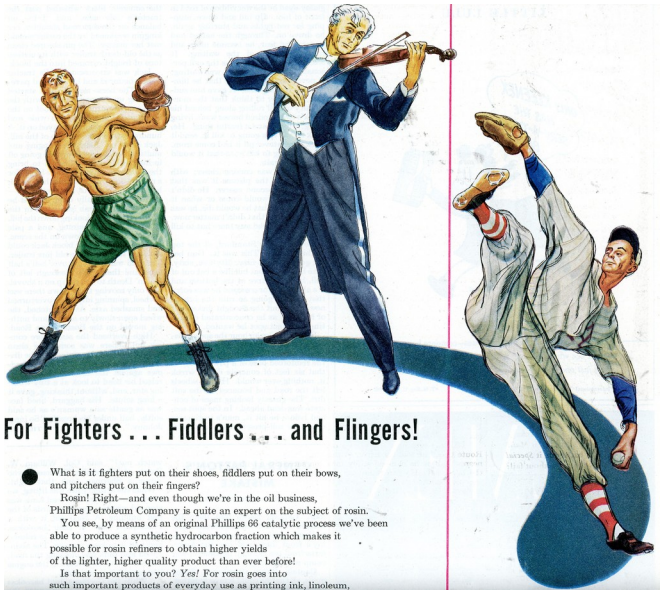

Figure 1: Image which has different labels such as person and human face

8 Conclusion

The idea of this Project was to learn and use the things taught in Course. My focus was to use the knowledge of Neural Networks for multiclass classification(training and testing). For that, I had to understand few concepts such as hidden layers, activation functions, optimization methods etc and also got to play with few parameters like number of neurons,layers,batch size. Main thing which I learned is building the Neural Network. Apart from that, I had to learn how to deal with Images in R and also to create databases for the dataset and read from it instead of reading it from file. In this way, through this project, I learned how to use TACC resources for database. Though I also wanted to use it for running the code but could not use it as I was not successful installing Keras on TACC machine(Submitted ticket but did not get any reply). As stated in the proposal, my intention was to combine both PostgreSQL and R, which I was successful to some extent. As I said in my proposal, result of the project was to get classification result of the images obtained using Neural network model and I was successful in learning how to do it though the result is not that good.

9 Useful links

- A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks
- Undrestanding Convolutional Layers in Convolutional Neural Networks (CNNs)
- ReLU
- Overview of gradient descent algorithm