Machine Learning Engineer Nanodegree

Capstone Proposal

Mayank Bhatnagar Udacity
Apr 13, 2019

Proposal

Domain Background

This project is about 'Fine-Grained Categorization'. The technique is advance compared to object identification problems. We can use such techniques to recognize different species of flowers, birds or any other species. It has wider use in identifying and grouping objects and even evaluating whether we are finding a new species next existing or explored earlier.

Steps required for an appropriate model:

1. Collecting large set of base data and fine tune it for usage
2. Dividing it into training, validation and testing sets
3. Creating a good benchmark for the data
4. Creating a model using a suitable algorithm
5. Fine-tuning the model to improve accuracy
6. Put human intelligence to refine the model, and further improve accuracy
7. Test with real-life examples, and further tune your model

My Personal Interest

Photography is my hobby. Images excites me. I use semi-professional DSLR camera and have my collection of 1000s of images. Sometime in future, I may use my learning of fine-grained categorization to help categorize my own pictures.

Problem Statement

I am planning to use CNN to help identify different specifics of monkeys. My training and validation data set will help train the model. On trained model, I will use sample images, both valid and invalid, captured from internet to test the accuracy of the model.

This model can be used to classify different specifies of monkeys or even help identify if a new species of monkey is found which does not match with any of the existing ones.

Accuracy of the identification will help determine how good the model is doing.

Datasets and Inputs

I am planning to use Kaggle website to get data set required for this project.

I have explored the existing data. Data has two sections, one for testing and other for validation. Each folder contains 10 subfolders corresponding to 10 different monkey species. The data set consists of large number of image files with minimum size of 300 * 400 Pixels.

Details of different monkey species can be found at Wikipedia's monkey cladogram. Following is the mapping of the folder name and the species:

1. n0, alouatta_palliata
2. n1, erythrocebus_patas
3. n2, cacajao_calvus
4. n3, macaca_fuscata
5. n4, cebuella_pygmea
6. n5, cebus_capucinus
7. n6, mico_argentatus
8. n7, saimiri_sciureus
9. n8, aotus_nigriceps
10. n9, trachypithecus_johnii

The training data is required to train the model, and validation is to validate the accuracy of the model. Post that I am planning to get some random images from the internet to test algorithm.

https://www.kaggle.com/slothkong/10-monkey-species

Solution Statement

I have already downloaded the data set. Thus, in summary, I would be evaluating the data set for the size and variation. Will create the CNN model. Train and Validate the model. Finally test the model using additional images.

For my solution, I am planning to use Transfer learning and Data Augmentation. Transfer learning using Keras model, and data augmentation to increase the number of training / validation objects.

I have explained the model design in the design section.

Benchmark Model

There are various Keras pre-trained model available like VGG16, VGG19, and RESNET. I would be exploring few to understand which one help me reduce the training time and increase efficiency.

I will verify the model using accuracy and loss functions.

Evaluation Metrics

I would be tracking following metrics to measure the performance of the model.

Training Time: The model created should be efficient and run fast for the data of this model.

Training / Validation Accuracy: It is important that we avoid over-fitting and under-fitting of the model. If it over-fitting we would be biased and if it under-fitting model will not perform well. I would be measuring this as one of the key metrics.

Training / Validation Loss: Similar to training / validation accuracy, training / validation loss also helps to understand whether the model is over fitting or under fitting. If the training loss is much lower, than validation loss than the model is over-fitting. If both the values are close to each other that means the model is under fitting.

Project Design

I have divided my solution into following parts:

Training / Validation Data Set Creation

1. Take the complete data set and divide it into training and validation data set
2. Per need, increase our image numbers by data-augmentation. Following could be done:
    a. Rotate image at a defined angle
    b. Shift the image to the side or up
    c. Zoom-in or Zoom-out images

Create CNN Model

3. Create a CNN model with initial size of 300 * 400. The final node size to be 10. CNN model would consist of following steps:
    a. Convolutional Layer with initial Input (300 * 400)
    b. Max Pooling to reduce special dimension
    c. Convolutional Layer
    d. Max Pooling
    e. Dropout
    f. Dense
    g. Softmax

    Reasons for using Convolutional Layer, Max Pooling Layer and Dense & Drop out Layer

    Convolutional Layer → Increases depth of the image: Using this we can increase the depth of the images say from 16 to 32 and then 32 to 64. This keeping the size of the image same. We can even use Relu activation layer to increase the efficiency by losing little accuracy.

Max Pooling Layer → Reduces the special dimension. After each convolutional layer, we will have max-pooling layer to decrease the image size.

Dense and Dropout Layer → Implement Dense layer to modify dimensionality, and in between added dropout layers to minimize overfitting. Finally, use 'Softmax' activation to convert the model into probabilistic values.

Train Model

4. Train the model using original and new set of images (if needed based on accuracy and loss values)
5. Measure the Training / Validation accuracy and loss to understanding the performance of the model
6. Modify CNN parameters (Layers, Node Levels, Dropout layers) to fine-tune the model.

Test the Model

7. Test the algorithm using test images. Use some valid pre-classified imaged and some invalid pre-classified images to get the right results.
8. Refine the model further if receive inaccurate results.

References

Fine-Grained Categorization:	https://vision.cornell.edu/se3/fine-grained-categorization/
Wikipedia:	Wikipedia's monkey cladogram.
Project from Kaggle:	https://www.kaggle.com/slothkong/10-monkey-species
Keras Documentation:	https://keras.io/
Others:	Many internet websites

Before submitting your proposal, ask yourself. . .

- Does the proposal you have written follow a well-organized structure similar to that of the project template?
- Is each section (particularly **Solution Statement** and **Project Design**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
- Would the intended audience of your project be able to understand your proposal?
- Have you properly proofread your proposal to assure there are minimal grammatical and spelling mistakes?
- Are all the resources used for this project correctly cited and referenced?