Capstone Project Proposal

Lok Chan

10/29/2017

Domain Background

While it is known that Convolution Neural Network does exceptionally well in classifying photographs, photographs are only a small subset of visual information that human encounter in real world. As a matter of fact, it is plausible to think that throughout human history, representing objects in photo-realistic details is a rather recent human phenomenon, as human beings have utilized visual representation of objects using drawing long before the invention of photographs. Nevertheless, identifying hand-drawn pictures embodies a different set of challenges, for it often contains less information than a feature rich photograph with colors.

It is no wonder, then, many recent studies in image classification focuses on identifying sketches made by human. The following academic works have shown success in this area:

Eitz, Mathias & Hays, James & Alexa, Marc. (2012). How do humans sketch objects?. ACM Transactions on Graphics. 31. 1-10. 10.1145/2185520.2335395.

Hand drawn sketch classification using convolutional neural networks,

www.iioab.org/articles/IIOABJ_7.S5_337-341.pdf

G. T. Sadouk, Lamyaa and E. H. Essoufi. A novel approach of deep convolutional neural networks for sketch recognition. international Conference on Hybrid Intelligent Systems, 2016

R. G. Schneider and T. Tuytelaars. Sketch classification and classification-driven analysis using fisher vectors. ACM Transactions on Graphics (TOG), 33(6):174, 2014.

Zhang, H, Liu S, Zhang C, Ren W, Wang, R, Cao X. [2016], SketchNet: Sketch Classification With Web Images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1105

Problem Statement

While sketches are often easily disernible to human eyes and producible, they are vey different than the pictures that are used to train popular CNN models. They lack the richness of the information found in photo-realistic representation, and they lack the predictable structure of words and numbers. Thus, one problem to be solved is to determine the appropriate input shape for model. While $224 \times 224 \times 3$ matrices seem to do well for feature-rich images, for black and white sketches this might be overkill. The next problem to be solve is to determine the number of hidden layers and their relevant parameters. The output layer will ultimately be a 250 node fully connected layer with soft-max activation function, producing a vector of the relative probabilities of the 250 object categories.

Data and Input

I will use the data-set available in "How Do Human Sketch Objects?", the webpage for the human sketch recognition study cited above with the same name. The data set consists of 500MB of png files.

There are in total 20,000 images available, divided into 250 categories by their objects. The images are deliberately lacking in features - for instance, objects are not to be situated in an environment, so there is no context that would help identifying the object. Further, they are also lack black areas that help defining the objects. According to the original paper, the taxonomy was the result of a crowd-sourcing effort. They then enlisted users from Amazon Mechanical Turks to draw sketches based on those categories.

Solution Statement

The solution is to construct a neural network that takes in the images as matrices (that will be smaller than $244 \times 244 \times 3$) for input, through a multiple Convolutional and Pooling layers, and ultimately generates a prediction through a 250 nodes output layers - there are 250 because there are 250 object categories.

Benchmark Model

For a benchmark model, I have constructed and tested a baseline CNN model. After I loaded the data-set and I took 7000 sketches as the training set, 5000 as the validation set, and 5000 as the testing set. I was unable to use all 20,000 due to a memory problem.

I constructed a baseline CNN model as follows:

```
from keras.layers import Conv2D, MaxPooling2D, GlobalAveragePooling2D
from keras.layers import Dropout, Flatten, Dense
from keras.models import Sequential

model = Sequential()
model.add(Conv2D(filters=16,kernel_size=4,
                padding='same',activation='relu',
                input_shape=(224,224,3)))
model.add(MaxPooling2D(pool_size=2))
model.add(GlobalAveragePooling2D())
model.add(Dense(250,activation='softmax'))
model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(train_tensors, y_train,
          validation_data=(valid_tensors, y_test[0:5000]),
          epochs=epochs, batch_size=20, verbose=1)
```

Evaluation Metrics

For this project, I will use cross entropy as a loss function during training. This seems appropriate since cross-entropy is low when the predicted probability is close to the actual label. Mathematically, cross entropy is defined as:

$$L(p, q) = -\sum_i p(x_i) log[q(x_i)]$$

where p is the true distribution of the data and q is the estimation of the model.

Since the goal is to have a classifier that correctly identifies the object category of a sketch, as an evaluation metric, I will use accuracy of the classification, which is defined as:

$$\frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

Using these metrics, the baseline model has obtained the accuracy of 0.3001% after 5 epochs. Since this is merely a baseline, my object is to do much better - I hope to achieve at least 50% accuracy.

Project Design

While hand-drawn sketches are most likely the first kind of visual representation that human beings have encountered, it depends on an impressive ability to extrapolate information from what basically is a group of lines. This implies the capacity to not only detect the physical features of a picture, but also make appropriate inferences by treating those features as an abstraction of a much more complex object.

A Deep Convolutional Neural Network is a powerful neural network algorithm that as achieve much success in image recognition. It functions by creating filters that are adapted to the features of the images the network receives as training inputs. They are then combined to draw probabilistic inference about pictures they received. The result is a powerful classifer that works well even in high dimensions.

In this project, I will try to develop a deep CNN that will have a reasonable degree of accuracy of sketch image classification. The basic workflow of the project is to improve the baseline model by experimenting with additional layers and its parameters. As with the benchmark model, I will split the data into the training, validation, and testing sets. However, I intend on using additional convolutional layers for the actual model. I also would add a drop-out layer to prevent over-fitting.