ImageNet Object Classification

Advances in Data Science Project

Hiren Patel, Vishal Diyora

Abstract:

We propose a deep convolutional neural network architecture that is based on the Google's Inception model that can give a new model for classification in the ImageNet Large-Scale Visual Recognition Dataset. Here we are exploring ways to scale up networks in ways that aim at utilizing the added computation as efficiently as possible by suitably factorized convolutions and aggressive regularization. We are trying to fine-tune the GoogLeNet Inception model with a different number of layers and also we are using different convolution methods those are used in original Inception model. We have used Keras and TensorFlow as backend.

Introduction:

In the last few years, mainly due to the advances in deep learning, more concretely convolutional networks, the quality of image recognition and object detection has been progressing at a dramatic pace. One empowering news is that the vast majority of this advance is not only the aftereffect of all the more capable equipment, bigger datasets, and greater models, however mostly an outcome of new thoughts, calculations, and enhanced system designs. The biggest gains in object-detection have not come from the utilization of deep networks alone or bigger models, but from the synergy of deep architectures and classical computer vision, like the R-CNN algorithm. The computational cost of Inception is likewise much lower than VGGNet or its higher performing successors. This has made it feasible to utilize Inception networks in big-data scenarios, where a huge amount of data needed to be processed at reasonable cost or scenarios where memory or computational capacity is inherently limited, for example in mobile vision settings. In any case, the many-sided quality of Inception design makes it harder to roll out improvements to the system. In any case, one should be wary about doing as such, as some directing standards ought to be seen to keep up high caliber of the models.

Code of Documentation:

We are using the portion (Dog images) of the ImageNet dataset instead of the whole ImageNet dataset because training the whole dataset will require a great amount of the resources, computational power and also it will take more time to train and test.

Dataset: http://vision.stanford.edu/aditya86/ImageNetDogs/

The code can be found with jupyter notebook:
https://github.com/vishal6557/ADS/blob/master/Final-Project.ipynb

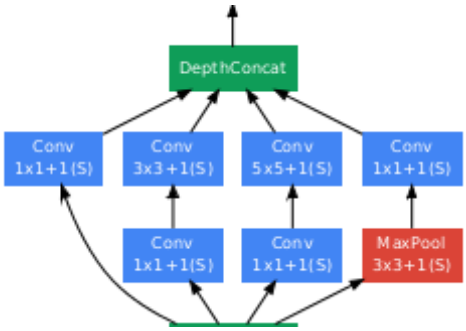
Methods:

The most straightforward way of improving the performance of deep neural networks is by increasing their size. This includes both increasing the depth – the number of levels – of the network and its width: the number of units at each level. This is as an easy and safe way of training higher quality models, especially given the availability of a large amount of labeled training data. However, this simple solution comes with two major drawbacks. Bigger size typically means a larger number of parameters, which makes the enlarged network more prone to overfitting, especially if the number of labeled examples in the training set is limited. This can become a major bottleneck since the creation of high-quality training sets can be tricky and expensive, especially if expert human raters are necessary to distinguish between fine-grained visual categories like those in ImageNet (even in the 1000-class ILSVRC subset).

Another drawback of uniformly increased network size is the dramatically increased use of computational resources. For example, in a deep vision network, if two convolutional layers are chained, any uniform increase in the number of their filters results in a quadratic increase of computation. If the added capacity is used inefficiently (for example, if most weights end up to be close to zero), then a lot of computation is wasted. Since in practice the computational budget is always finite, an efficient distribution of computing resources is preferred to an indiscriminate increase of size, even when the main objective is to increase the quality of results.

Our Model:

The main idea of the Inception architecture is based on finding out how an optimal local sparse structure in a convolutional vision network can be approximated and covered by readily available dense components. Note that assuming translation invariance means that our network will be built from convolutional building blocks. We have reduced the number of layers in the inception model.
Our model starts with a sequential chain of convolution, pooling, and local response normalization operations, in a similar fashion to previous convolutional neural network models, such as AlexNet. Later papers on the inception architectures refer to this initial segment as the 'stem'.



The above diagram shows an inception module. GoogLeNet contains nine of these modules, sequentially stacked, with two max-pooling layers along the way to reduce the spatial dimensions.

Results/Conclusion:

We are trying to run our model and got validation accuracy as 30% with training accuracy as 95%. Our model is getting overfitted so, we are trying to change the architecture layer to reduce the overfitting of the model.

Future Work:

We have trained our model with the portion of the ImageNet image dataset due to time constraints. We will train our model with the ImageNet Large Scale Visual Recognition (ILSVR) and will try to improve our model's accuracy by changing in the hyperparameters and the number of layers if it gives more accuracy.

References:

1. GoogleNet in Keras: http://joelouismarino.github.io/blog_posts/blog_googlenet_keras.html
2. Going Deeper with Convolutions: https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf
3. Keras Documentation: https://keras.io/
4. TensorFlow Image Recognition Tutorial: https://www.tensorflow.org/tutorials/image_recognition
5. Convolution Neural Networks for Visual Recognition: http://cs231n.github.io/convolutional-networks/
6. Dog breed classification with Keras: http://machinememos.com/python/keras/artificial%20intelligence/machine%20learning/transfer%20learning/dog%20breed/neural%20networks/convolutional%20neural%20network/tensorflow/image%20classification/imagenet/2017/07/11/dog-breed-image-classification.html
7. How convolution neural network Works: https://www.youtube.com/watch?v=FmpDIaiMIeA&t=634s