Review of "ImageNet Classification with Deep Convolutional Neural Networks"

Riyad Bin Rafiq

1. Paper summary

The paper [1] implemented a deep convolutional neural network (CNN) using a high-resolution ImageNet dataset to classify 1000 categories. The convolutional neural network included five convolutional layers and three fully connected layers. Moreover, 60 million parameters were trained and to avoid the overfitting, the authors applied some techniques, such as data augmentation and dropout. A large network with a huge number of parameters usually takes a longer period of time for training, so, making the training faster, the authors applied an efficient GPU implementation for the convolution operations. Experimental results showed that they obtained top-1 and top-5 error rates of 37.5% and 17.0% in classification during the ImageNet LSVRC-2010 competition. Furthermore, the authors applied a variant of the model in ILSVRC-2012 contest and achieved top-5 error rate of 15.3% while the second-best error rate was 26.2%

2. Contribution

2.1 Applied larger dataset

At the time of the research, small datasets of labeled images were available. So a simple object recognition task was easy to perform using machine learning methods. But recognizing objects in realistic settings with some variations needs a dataset of millions of images. In this research, 1.2 million images from the popular ImageNet dataset (consists of 15 million images) were used for training the model.

2.2 A unique CNN with preventing overfitting

Training a model using millions of images needs a large learning capacity. Feedforward networks are hard to train as the network needs to deal with a massive number of parameters and takes longer training time compared to CNN. In the research, the authors proposed a unique deep CNN with preventing overfitting techniques such as dropout and image augmentation to recognize 100 classes. This inspired a lot of researchers to develop a CNN architecture to solve various image classification tasks.

2.3 GPU parallelization

A CNN architecture is expensive when it is applied in large scale high-resolution images. 1.2 million images are too big to fit on one GPU. The authors applied GPU parallelization and spread the net across two GPUs to compute convolution operations, and so, half of the kernels (neurons) were employed on each GPU. Moreover, they made the implementation publicly available for the community [2].

2.4 Feasibility of ReLU Nonlinearity

In the research, the authors demonstrated that ReLUs train faster than it's equivalen tanh when deep CNN is applied. For example, they took the CIFAR-10 dataset and showed that a four-layer CNN with ReLUs reached 25% error rate six times faster than a similar size network with tanh.

3. Critique

3.1 Removing convolutional layer

The authors stated that removing any convolutional layer decreased the model's performance. Also, they mentioned that removing any of the middle layers resulted in 2% loss for the top-1 performance. As there are five convolutional layers in the network, in my opinion, it would be better if they provided the loss of performance for removing each of the convolutional layers. Because we know that early layers learn general or low-level features while the later layers learn more complicated/detailed features of the object. By doing so, the research represents a robust intuition about the importance of the depth and how each convolutional layer affects the modeling strategy.

3.2 Why dropout 0.5?

Dropout 0.5 was applied to prevent the overfitting while building the network architecture. But they didn't mention why the specific value, 0.5, was chosen. Were there any other dropout values they applied rather than 0.5? How did those values perform in the model's performance? A lot of questions emerge.

3.3 Details of learning

The authors trained their model using stochastic gradient descent with a batch size of 128, momentum of 0.9, weight decay of 0.0005 and learning rate of 0.01. But they only explained the reason for choosing the values, weight decay and learning rate. But how they chose batch size and momentum is missing. They didn't mention whether they chose those randomly or by applying any grid search.

Reference

1. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. 2012;25. Available: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

2. Code. [cited 7 Mar 2022]. Available: https://code.google.com/archive/p/cuda-convnet/