Rethinking the Inception Architecture for Computer Vision

Research paper by Christian Szegedy ; Vincent Vanhoucke ; Sergey Ioffe ; Jon Shlens ; Zbigniew Wojna.
Presentation by Emily Da

Summary

Convolutional networks are filters that detect patterns and have several layers known as convolutional layers. They are the foundations for most computer vision solutions that complete tasks such as image and video recognition, medical image analysis, financial time series, analyzing documents and many other tasks related to computer vision. Ever since AlexNet (a convolutional neural network or CNN) won an ImageNet competition in 2012, convolutional networks have become more recognized in the world of computer vision. Increasing the model size and computation costs translates to an immediate quality gain, however, computational efficiency and the parameter count is something that still needs to be worked on. From 2014, the quality of network architectures started to improve significantly and deeper networks are able to be utilized. A CNN architecture called Inception was developed by Szegedy et al. and this paper will be discussing this. The aim of this paper is to explore ways to scale up networks, how to increase the model size efficiently, with a minimal computational cost increase.

Key points:

- **Factorising:** A suggestion to create sparsely connected convolutional architecture rather than a fully connected one. For example: to break a 5x5 layer into smaller layers of 1xn and 1xn. This is a way to increase network size without dramatically increasing the computational cost.
- **Reducing grid size significantly:** We can expand the network filters but the computational cost becomes expensive. We can use a method called pooling, but that creates a bottleneck, which limits the flow, slowing down the process.
- **Minimising computation cost:** Simply increasing filter bank sizes will increase the computation costs disproportionately. Decreasing the parameters will decrease the cost, but how do we keep the computation cost low while improving the efficiency?

Discussion questions

1. Is factorising and increasing the number of filters to two or multiple nx1 layers really better? Why haven't other CNNs implemented this if so?
2. What are some ways that could scale up the networks?
3. Do the pros outweigh the cons?