Deep Neural Networks for Mobile Platforms

Oleksandr Baiev

PhD, Sr. Engineer
Samsung R&D Instutute Ukraine

AI Ukraine, 2016

Outline

Problems overview
- Key problems
- Deeper problem overview

Topology
- Overview

Quantization
- Overview

Pruning
- Overview
- Examples

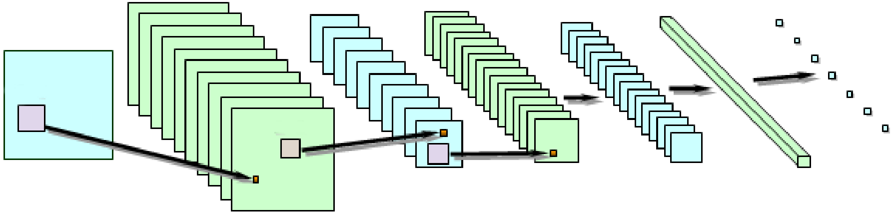
Outline

Neural networks

- Cutting edge results for CV, NLP, Signal Processing, Recommendations.
- Unified solution for different problems.

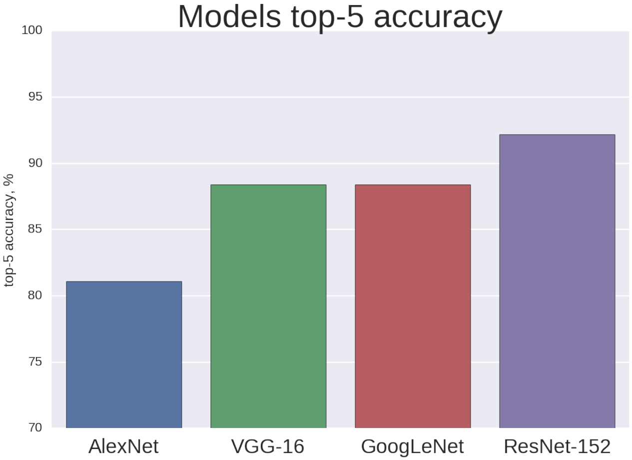
Neural networks

Convolution neural network



Deployment stage problem

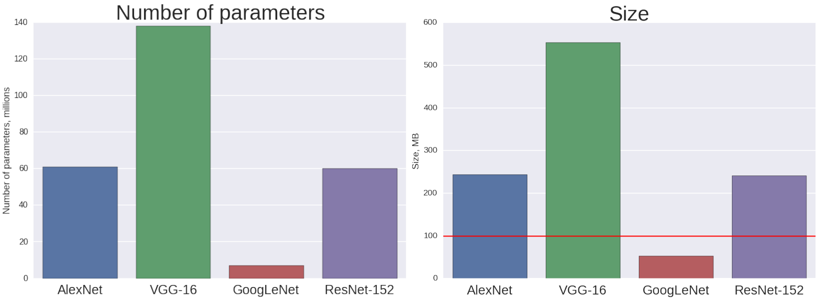
Models quality



- Alex Krizhevsky, et al. ImageNet Classification with Deep Convolutional Neural Networks, 2012
- Christian Szegedy, et al. Going Deeper with Convolutions, 2014
- K. Simonyan, et al. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014
- Kaiming He, et al. Deep Residual Learning for Image Recognition, 2015

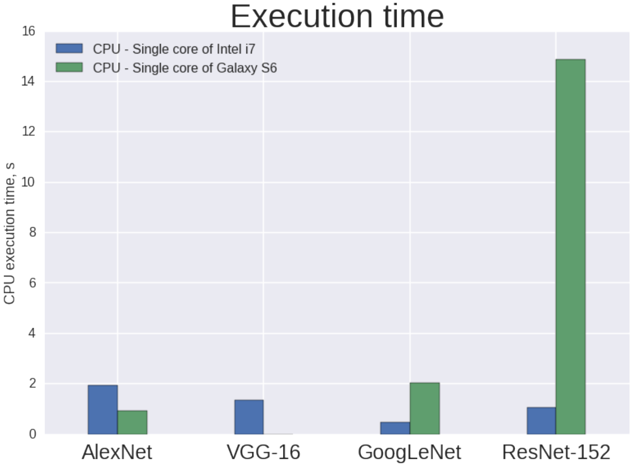
Deployment stage problem

Models size



Application more than 100MB requires WiFi for downloading via app stores

Deployment stage problem
Models execution

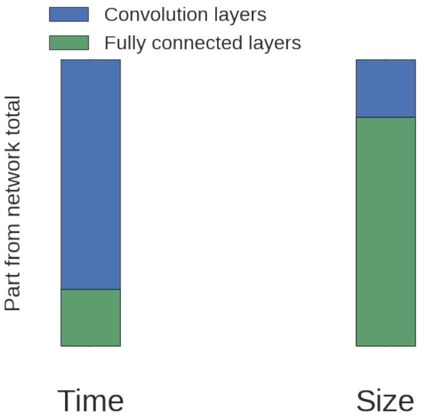


Test with https://github.com/sh1r0/caffe-android-lib

Outline

Problem overview

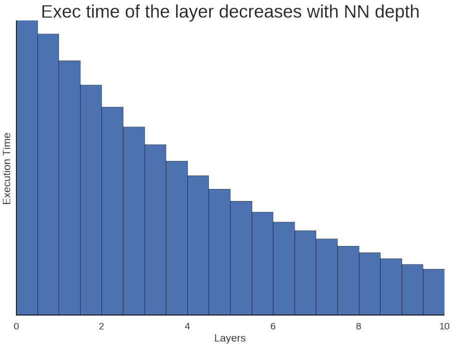
Approximate layers distribution

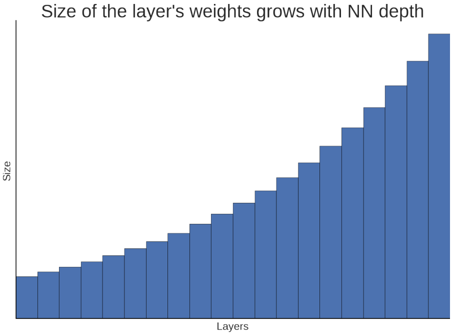


- Fully Connected are bigger than Convolution layers in terms of MB
- Convolution takes much more time for forward pass
- Target device have to store layer's feature maps in RAM for at least one layer

Problem overview

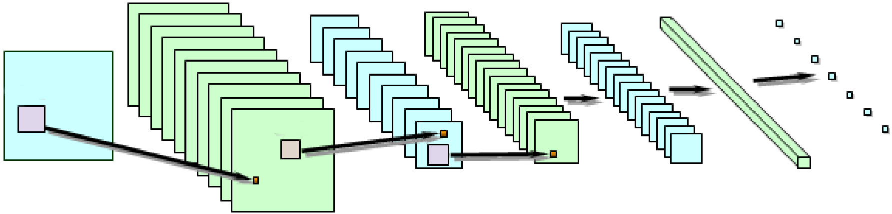
Execution time and size by layer





Problem overview

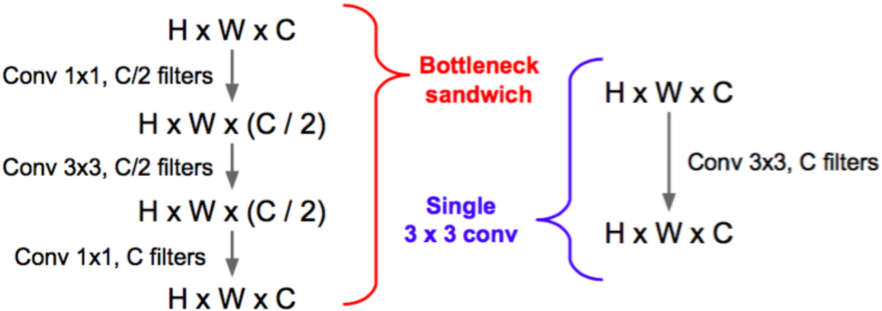
Parameters importance



- Feature map's **width, height** influence on execution time
- Feature map's **depth** influences on model size

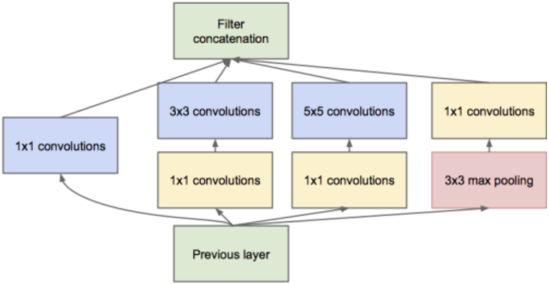
Outline

Bottlenecks



- Min Lin, et al. Network In Network, 2013

Inception module

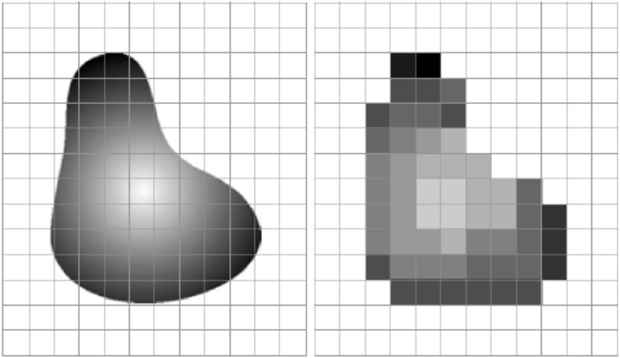


Compose different kernel sizes

- Christian Szegedy, et. al., Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, 2016
- Christian Szegedy, et. al., Rethinking the Inception Architecture for Computer Vision, 2015

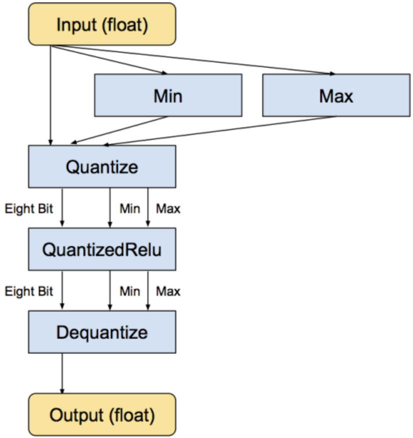
Outline

Quantization



- Pete Warden, How to Quantize Neural Networks with TensorFlow, 2016
- Matthieu Courbariaux, et. al., BinaryConnect: Training Deep Neural Networks with binary weights during propagations, 2015

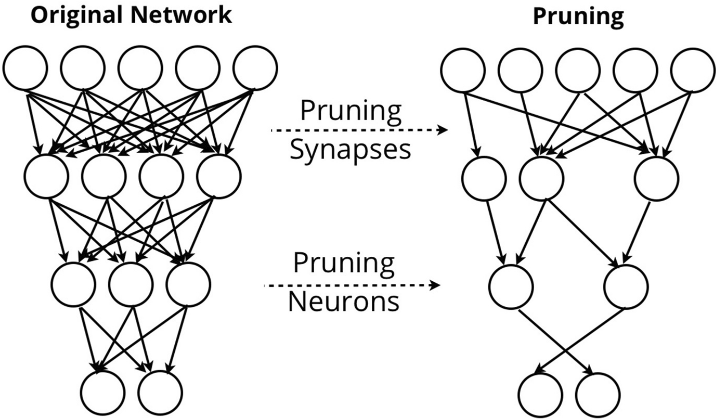
Quantization Schema



- All operations use precalculated Min and Max values which are used for rescaling
- Min and Max values selected from function behavior and real number values

Outline

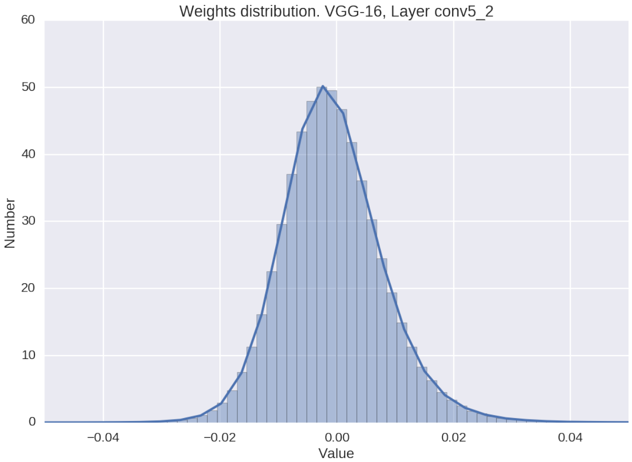
Pruning basics



The idea of pruning is removing unimportant weights. The one question is how to define "unimportant".

Pruning basics

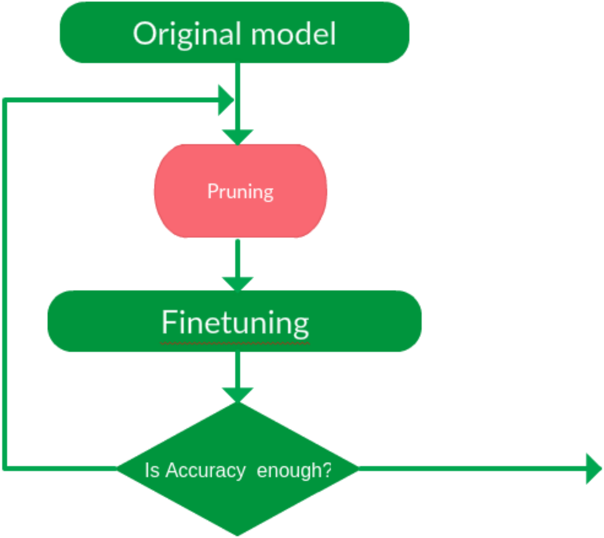
Unimportant criteria



- Yann Le Cun, et al. Optimal Brain Damage, 1990
- Babak Hassibi, et al. Second Order Derivatives for Network Pruning: Optimal Brain Surgeon, 1992
- Song Han, et al., Learning both Weights and Connections for Efficient Neural Networks, 2015

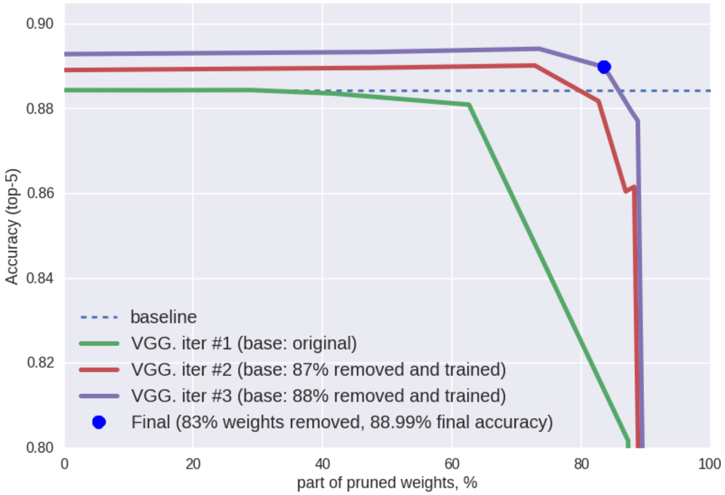
Pruning basics

Iterative Process



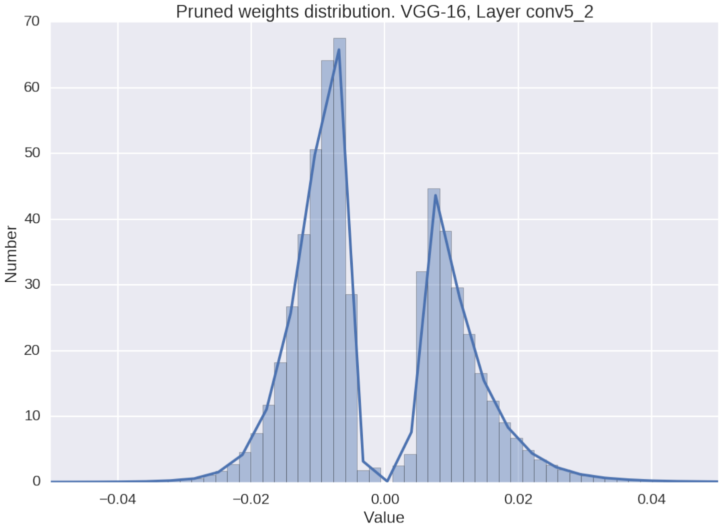
Outline

Pruning example VGG

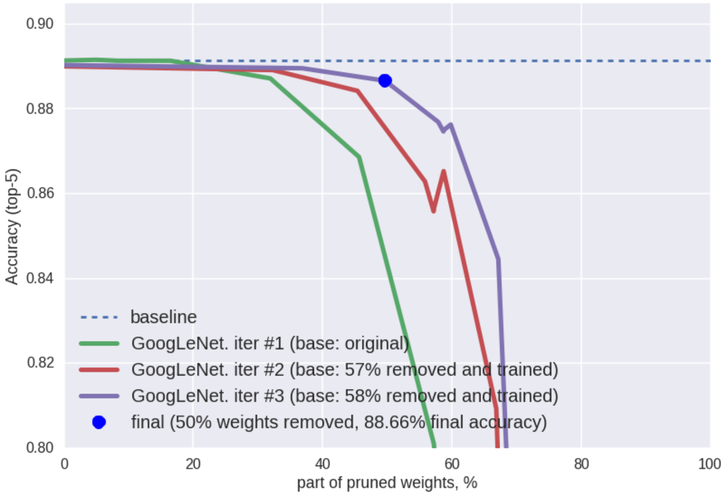


- Song Han, DSD: Regularizing Deep Neural Networks with Dense-Sparse-Dense Training Flow, 2016

Pruning example VGG

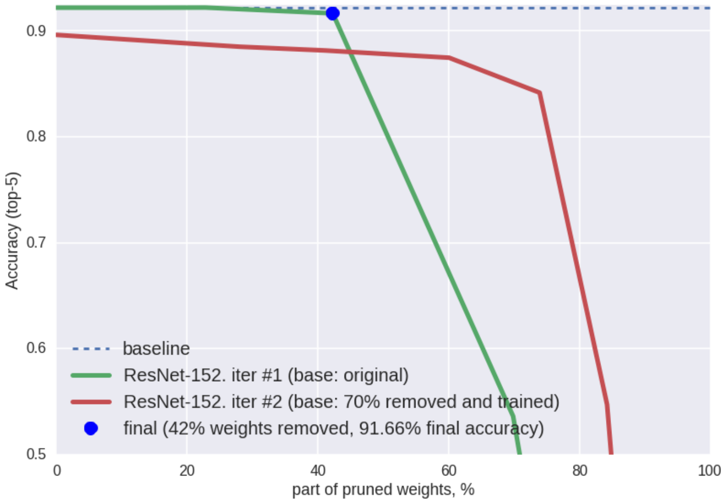


Pruning example GoogLeNet



- Hao Li, et.al., Pruning Filters for Efficient ConvNets, 2016

Pruning example ResNet-152



Summary

- Deep neural networks provides excellent quality, but requires powerful computation instances
- There are several simple and useful approaches for reducing required memory size and execution time without increasing hardware cost