ILSVRC Winning Model Weights

Aramis Subee

July 5, 2019

1 Introduction

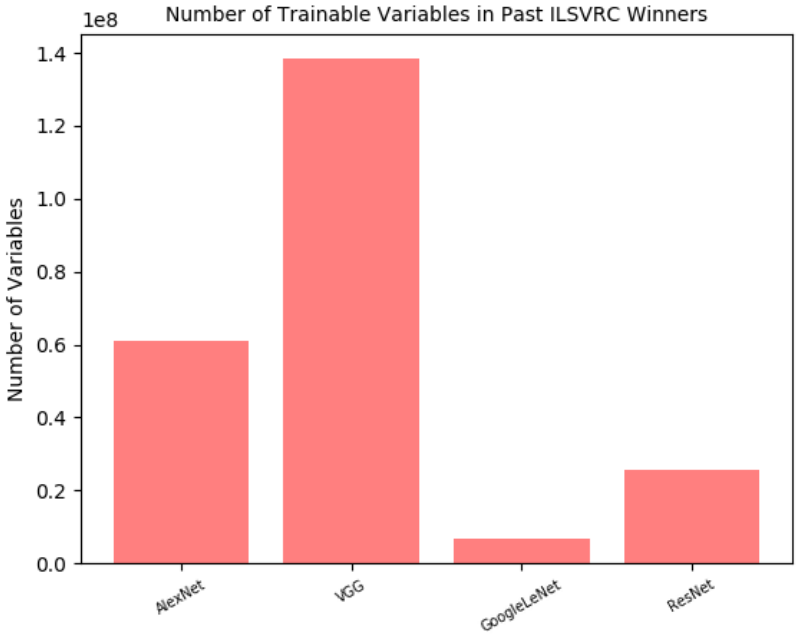
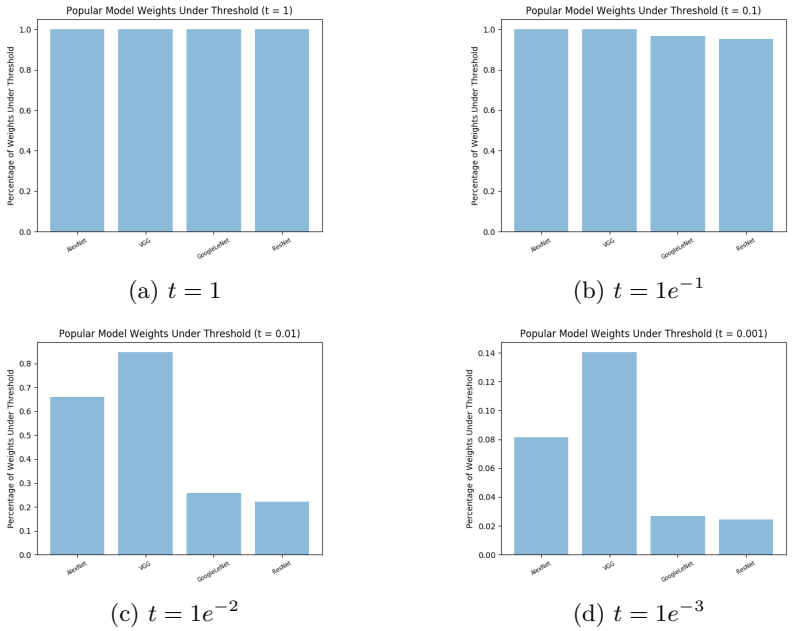


Figure 1: Size of most successful models in ILSVRC competition from 2012-15.

ILSVRC is the premier Computer Vision competition, testing Machine Learning algorithms on their ability to detect objects and classify images. As computational power increases, the most successful models are becoming larger in their network size. Recent successful models have all had network sizes on the magnitude of 10^7 trainable variables, with VGG even reaching 10^8 variables (Figure 1). There has been extensive work into the benefit of increasing the number of layers in networks, and therefore also the number of variables in most cases.



(a) $t = 1$ (b) $t = 1e^{-1}$

(c) $t = 1e^{-2}$ (d) $t = 1e^{-3}$

Figure 2: Percentage of variables in ILSVRC winners under various threshold values.

But as these network sizes increase, storage becomes a limiting factor, incentivizing sparser architectures. This report explores the feasibility of sparsifying the successful ILSVRC models. We do so by exploring the percentage of weights that fall under a certain threshold in each model.

2 Results

The results of our experiment are shown in Figure 2. Pretrained models for AlexNet, VGG, and GoogLeNet are taken from PyTorch. The pretrained model for ResNet was taken from Keras Applications.

All variables in all models fall under the $t = 1$ threshold. Even all variables in AlexNet and VGG fall under the $t = 1e^{-1}$ threshold, with an overwhelming majority of GoogLeNet's and ResNet's weights falling under the same threshold. It's only when t is lowered further do we find where the majority of the variable values fall. Even at the lowest threshold value, a significant fraction of the variables in all models still fall within the threshold bounds. These variables take on such small values that one wonders if they really have an impact in the grand scheme of the network. Perhaps they present an opportunity to sparsify the networks, maybe even zeroing out the smaller variables and testing network performance.