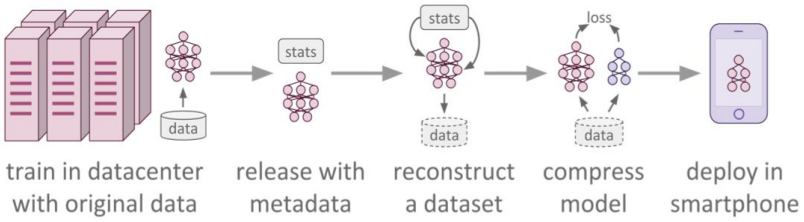
Data-free Knowledge Distillation for Deep Neural Networks

Introduction

Many efficient model compression techniques have been proposed recently, but all of these approaches rely on access to the original training dataset. Now, this access might not always be possible if the network to be compressed was trained on a very large dataset, or on a dataset whose release poses privacy or safety concerns. This paper proposes a method for data-free knowledge distillation, which is able to compress deep neural networks trained on large-scale datasets leveraging only some extra metadata to be provided with a pretrained model release.



Method

After training the teacher network on the original dataset, the authors compute records for the activations of each layer in the network, and save those alongside the model. Now in order to train the student without access to the original data, they attempt to reconstruct the original dataset using only the teacher model and its metadata in the form of precomputed activation records. How? First, pass random gaussian noise as input to the teacher model, then apply gradients to that input noise to minimize the difference between the activation records and those for the noise image. Doing this repeatedly allows to partially reconstruct the teacher model's view of its original training set.

Mathematical Formulation

Given a neural network representation ϕ, and an initial network activation or proxy thereof $\phi_0 = \phi(x_0)$, the goal is to find the image x^* of width W and height H that:

$$x^* = arg\,min_{x \in R^{H \times W}} l(\phi(x), \phi_0)$$

where l is some loss function that compares the image representation $\phi(x)$ to the target one ϕ_0.

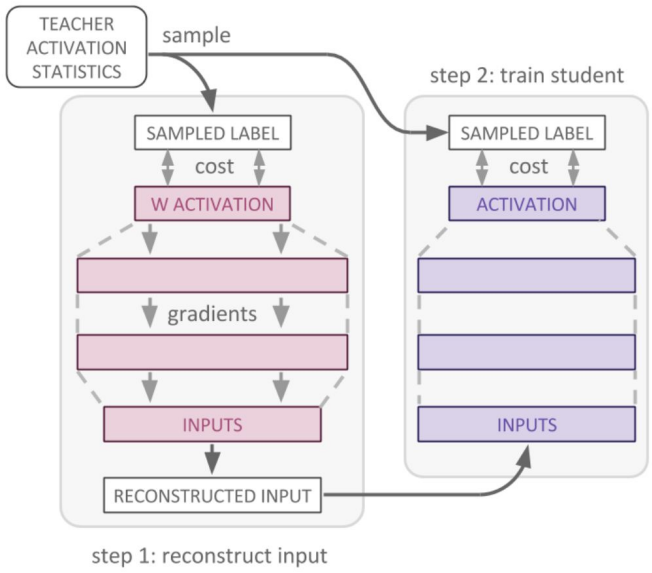
Activation Records

1) Top Layer Activation Statistics

The simplest activation records are the means and covariance matrices for each unit of the teacher's classification layer. Record these statistics according to the below equations:

$$\mu_i = Mean(L_i/T), Ch_i = Chol(Cov(L_i/T))$$

where L refers to the values in the network right before the final softmax activation, i refers to the i^{th} unit in that top layer, and T refers to some temperature scaling parameter. To reconstruct the input, sample from these statistics and apply ReLU to it. Then replace the student's topmost non-linearity with ReLU and minimise MSE loss between these two activations by optimizing the input of the network, thus reconstructing an input that recreates the sampled activations.



2) All Layers Activation Statistics

Unfortunately the above method is underconstrained: there are many different inputs that can lead to the same top-layer activations. To better constrain the reconstructions, store records for all layers instead of just the top-most. The reconstruction procedure is the same as the one above, except that for hidden layers the statistics are described as follows:

$$\mu_i = Mean(L_i), Ch_i = Chol(Cov(L_i))$$

The optimisation objective used is the sum of the MSE for each layer, normalized by the number of hidden units in the layer.