We have shown there is a relatively small amount of information that, if added to a release of a pre-trained model, can facilitate network compression. We have also shown that it is possible to compress a network with no access to the original training set and have motivated further exploration into distribution formats for deep models.

**Knowledge Distillation** [2]: training a student model on the weighted activations of a teacher model on the train set. Input reconstruction works best when statistics for all layers (rather than just top layer) are stored and optimized for: per-pixel means.

Why not train the smaller architecture directly? Training bigger deep learning models leads to better accuracies, due to techniques like dropout, which enables generalization. Smaller models can theoretically learn these functions [1], but training is hard.

Knowledge Distillation [2]: training a student model on the weighted activations of a teacher model on the train set.

As datasets get larger, their release becomes prohibitively expensive. Even when a big dataset is released [3], it usually represents a small subset of a much larger internal dataset, used to train many of state of the art models.

**Problem:** is there metadata can be provided with a pre-trained model to enable more efficient compression, even when no training data is available?

**Idea:** keep per-layer activation statistics for the teacher model. Reconstruct an input that matches those statistics using Gradient Descent. Inspired by Draelos et al. [4].

Since the input to the student was reconstructed from the teacher, it should ideally provide even more information about how the teacher generalizes.

**REFERENCES**


