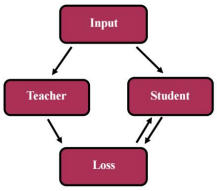


Exploring Effects of Knowledge Distillation in Model Compression and Accuracy

Authors: Matthew Tan, Evan Sheehan, Dian Ang Yap
Stanford University

Introduction

Current state of the art deep learning models for image recognition algorithms often utilize large and deep networks. Training such models often require multiple GPUs over an extended periods of times. However, recent research by Song Han (SqueezeNet), Bengio (Fitnets) and Hinton (Knowledge Distillation) has proven the potential of creating much smaller networks without substantial loss in accuracy. In some cases, these networks can be further post-processed to match / exceed the larger network's performance.

Our goal in this project was to delve into Knowledge Distillation and evaluate its feasibility, accuracy and effect on smaller networks. We tested this model on MNIST to check its feasibility and then on a larger dataset to prove its scalability. To do so, we retrofitted MobileNet, a state of the art image classifier, to the Cat Dog dataset, pruned it, and used Knowledge Distillation with the pruned model. Doing so allowed us to discover the effects of Knowledge Distillation on various models.

Project Overview

Dataset: We used MNIST to test this architecture's feasibility and the Kaggle Cat Dog Dataset to test scalability. We used an 80-20 split for train & test sets.

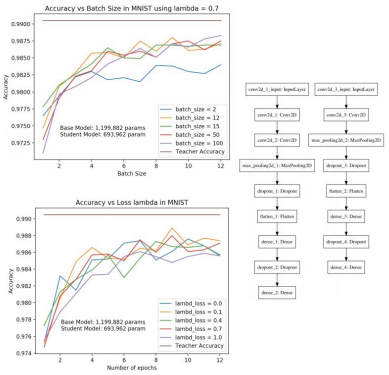
Model: For MNIST, we used state of the art models from the open source community. For the Cat Dog Dataset, we utilized MobileNet with transfer learning applied to it. Both of these were compressed to create the teacher-student architectures from which we trained our models.

Features: Pre-processing was done on the images (resizing, randomization) to fit them to the model being used.

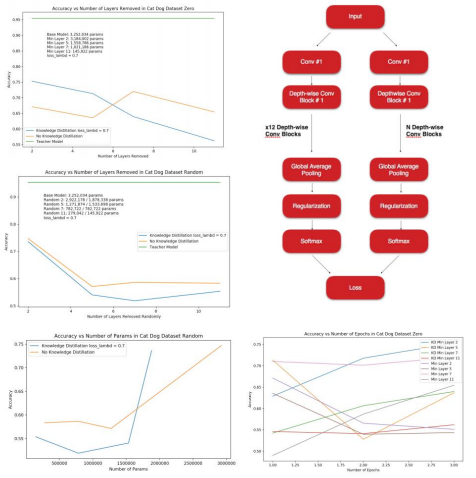
Method: To evaluate MobileNet, we first retrofitted the last layers in the teacher to match the Cat Dog Dataset and then applied transfer learning in order to fit it. We then removed layers from it, using various methods, to create the student, including zero density, random layers , etc., and evaluated different ways of weight initialization for it, including pre-loaded weights from teacher and randomly initialized weights. We then retrained the teacher-student architecture, keeping the teacher's weights frozen so that backpropagation was only carried out on the student. Various weightings of the teacher-student architecture were further explored, including varying the loss lambda parameter and tuning other hyperparameters (batch size, number of epochs, etc.).

Loss function $(1-\lambda)(y_{\text{true}} - y_{\text{pred}}) + \lambda * (y_{\text{teacher}} - y_{\text{pred}})$

Results



Using different lambda loss parameters can allow for faster convergence and even higher accuracy than simply learning from the ground truth. However fully trusting the teacher network seems to negatively affect the accuracy of the model.



Discussions

Results Discussions

We observe that knowledge distillation appears to work exceptionally well, compared to standard pruning, when the number of layers removed is decreased by only a small number amount, but seems to have detrimental effects when the proportion of layers removed from the teacher increases to a significant fraction of the overall size of the network.

Furthermore, we observe that certain random layers that are removed seem to possess more gravity in altering results when they are removed than others. We also see that knowledge distillation seems to cause converge at a much faster rate, or requires fewer epochs to converge, but also drops a lot faster as the number of layers removed begins to increase. We hypothesize that the number of epochs is not sufficient to draw any broader conclusions than this. Ideally, we would be running far more than 3 epochs with the full dataset to properly train the model, but GPU and cloud processing limitations hindered our ability to fully explore the space of possible avenues comprehensively.

Future Directions

Limited computational power has constrained the extensiveness of our results. However these results present a strong case for the possibility of compressing models without much loss of accuracy. Different methods of removing layers were seen to impact the accuracy of the model.
Running the model over the range of loss-lambdas for the cat-dog dataset may show us much more interesting details. Further, the cat dog dataset, while consisting of large images, is relatively small in size. Testing the model on larger datasets such as ImageNet may yield much more useful / interesting results.
Finally, a technique discussed in the related works but not explored in this was the use of hints and the controlling of both their position and number. In some ways, pre-loading the weights is some form of hint, although a possible extension would be to run it on randomly initialized weights and see what the effect of this would be.

References:
* Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the Knowledge in a Neural Network."
* Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta & Yoshua Bengio, "FITNETS: HINTS FOR THIN DEEP NETS."
* Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer, "SQUEEZENET: ALEXNET-LEVEL ACCURACY WITH 50X FEWER PARAMETERS AND <0.5MB MODEL SIZE."