APMTH 207: Advanced Scientific Computing:

Stochastic Methods for Data Analysis, Inference and Optimization

Group Project -- Milestone 1

Harvard University
Fall 2018
Instructors: Rahul Dave
Due Date: Tuesday, November 13th, 2018 at 11:59pm

Instructions:

- Upload your iPython notebook containing all work to Canvas.
- Structure your notebook and your work to maximize readability.

Tutorial on a Recent Research Development

As part of the requirements for this course we've stated the following(quoting the website): "There will be one paper, towards the end of this course. It will require reading and presenting a recent research development in the field."

We've compiled a list of papers that contain interesting treatments of topics related to AM207:

https://docs.google.com/document/d/1N0FjmMHfpX8P__TAzWLQYXqtj_Hbgo839Gu3B0oKOaM/edit?usp=sharing
(https://docs.google.com/document/d/1N0FjmMHfpX8P__TAzWLQYXqtj_Hbgo839Gu3B0oKOaM/edit?usp=sharing)

If you find a paper not on this list that you would like to delve into for your project, please let us know (and we can add it).

We want you to create a tutorial style jupyter notebook summarizing the relevant math, methods and procedure in a paper of your choice. Your tutorial should show the relevant methods in action by implementing them on appropriately chosen data.

At this point you should have organized yourself into a group, have chosen a paper topic and discussed it with a member of the AM207 teaching staff. In this milestone, please give us the name of your group(s) in Canvas, the name and url of your paper, a a short summary of your paper, a short description of your plan of attack, and the AM207 teaching staff member with whom you discussed (and from whom got approval of) your paper and plan of attack.

My/Our Paper (give us the title and url):

-- Paper Title: Distilling the Knowledge in a Neural Network

-- Paper Url: https://arxiv.org/abs/1503.02531 (https://arxiv.org/abs/1503.02531)

-- Brief Description:

Geoffrey Hinton, Oriol Vinyals, Jeff Dean

A very simple way to improve the performance of almost any machine learning algorithm is to train many different models on the same data and then to average their predictions. Unfortunately, making predictions using a whole ensemble of models is cumbersome and may be too computationally expensive to allow deployment to a large number of users, especially if the individual models are large neural nets. Caruana and his collaborators have shown that it is possible to compress the knowledge in an ensemble into a single model which is much easier to deploy and we develop this approach further using a different compression technique. We achieve some surprising results on MNIST and we show that we can significantly improve the acoustic model of a heavily used commercial system by distilling the knowledge in an ensemble of models into a single model. We also introduce a new type of ensemble composed of one or more full models and many specialist models which learn to distinguish fine-grained classes that the full models confuse. Unlike a mixture of experts, these specialist models can be trained rapidly and in parallel.

Members of the Group (give us the names and emails of all collaborators):

-- Collaborator 1: Jiejun Lu

-- Collaborator 2: Chia Chi (Michelle) Ho

-- Collaborator 3: Jiawen Tong

-- Collaborator 4:

Project Group(Name of Project Group in Canvas):

-- Project Group Name (FAS): PaperTutorial 4

-- Project Group Name (DCE):

Teaching Staff Member(s) we'd like to discuss our project with:

-- Teaching Staff Member 1: Patrick

-- Teaching Staff Member 2: Rahul

Summary:

Write a max 8-9 sentence summary of your paper

Training a powerful model that accurately extracts structure from data is often at odds with training a deployable model as the former objective tends to result in large and cumbersome models while the latter has much more stringent requirements on latency and computational resources. A way to overcome this conflict is to first train a standalone cumbersome (teacher) model with high accuracy then to transfer the learned knowledge to a smaller (student) model which will be deployed for real application usages. In general, knowledge transfer is done by using the class probabilities produced by the teacher model as “soft targets” for training the student model. The authors proposed the “distillation” approach: Raising the temperature of the final softmax until the teacher model produces a suitably soft set of targets and using the same high temperature when training the student model to match these soft targets. When the correct labels are known for all or some of the transfer dataset used to train the student model, the training can be enhanced by using a weighted average of two different objective functions: 1) the cross entropy with the soft targets computed at the same high temperature used to generate the targets from the teacher model; 2) the cross entropy with the correct labels computed at a temperature of 1. Using the MNIST dataset, the authors first showed that an unregularized network with two hidden layers of 800 rectified linear hidden units trained on the dataset directly performed worse than the same network architecture trained from a heavily regularized teacher network with two hidden layers of 1200 units at a temperature of 20. They further showed that, for an even smaller student network with two hidden layers of 30 units, the optimal temperature range for training is between 2.5 to 4. Finally, the authors showed that, with the right biases that optimizes test performance, 1) the student network can get 98.6% of the test digist 3s correct despite never having seen a 3 during training and and 2) the test error is only 13.2% even when the transfer set contains only 7s and 8s.

Plan of Attack:

Write a max 5-6 sentence description of your plan of attack for your project tutorial

We will demonstrate how increasing the temperature of the softmax layer generates “softer” targets for knowledge distillation. We plan to reproduce the paper’s MNIST result by 1) training a cumbersome teacher model with 2 hidden layers of 1200 ReLU activated units and regularized with dropout and weight constraints; 2) distilling the knowledge from the teacher network to a smaller student network with 2 hidden layers of 30 or 800 ReLU activated units without regularization. We will vary the temperature to determine the relationship between the complexity of the student network and temperature with regards to distilled model performance. We will also experiment with different bias values under different temperatures to understand the relationship/dependency between them. Finally, we will apply the method to a synthetic data set with varying degrees of separability in order to show the impact of class separability on the optimal training temperature and model bias for distillation.

I got Approval of my Plan of Attack from:
Patrick and Rahul
