ALPhA Summer Week 3 Presentation

BRADEN KRONHEIM

JUNE 21, 2019

Summary

- Put the Riemann Manifold HMC algorithm into the BNN code
- Initially it worked, but had an absurd amount of RAM usage
  - Around 35 GB for a network with 19 inputs, 1 hidden layer with 20 neurons, and 1 output value
  - The network trained a descent pace
- The way the code was written initially, the @tf.function decorators would not work without throwing errors, such as a matrix not invertible error
  - This error did not show up when running without the decorator
- Now, the code is able to run without throwing errors, but does not train properly when the decorators and there, and does train when they are not there

Problems from @tf.function and some solutions

- Matrix not invertible error:
  - This error was thrown with the decorator but not without it.
  - It appears that tf.linalg.det may be calculated differently in graph and in eager mode.
  - In order to combat this, I used a Singular Value Decomposition to obtain my determinant and matrix inverse along with tf.where() statements to combat singular matrices.
- Nan's from Gradient Tape:
  - Nan's were showing up for no good reason when I took the gradient of the loss function with respect to the network parameters.
  - Apparently, Gradient Tape follows both cases in tf.where() statements, so a "safe" calculation is necessary.
    - This means, for example, replacing values less than or equal to 0 for a log before taking it, instead of taking the log of two different values depending on the input value.

Current Problem

- The code currently runs without throwing errors, but depending on what functions have the @tf.function more problems can appear
  - Sometimes the NaN's come back in the gradient calculation
  - Sometimes Gradient Tape thinks the second derivatives do not exist
- In order for this algorithm to be feasible, this graph structure is required.
  - Instead of consuming around 40 GB of RAM, the networks will consume around 7 or 8.
  - Additionally, the code runs far faster in the graph structure

Goals for next week

- Try and make the RMHMC algorithm work
  - If I am unable to do this by the end of Monday, I will probably move on
- Perform data analysis on my networks from last semester
  - Analysis of error location
  - Predictions of regions with high cross section
- Run longer experiments with Prelu activation and Cauchy prirors
  - Also with the RMHMC algorithm should it start working