Project Summary:

I propose to explore neural networks and how they approximate functions. More specifically, I plan on looking into literature surrounding large limits of Bayesian networks. This allows for many different aspects to be tied together, such as: uncertainty estimation, theoretical convergence, test accuracy, and training/model speed. Overall, it's promising as a mix of theory and practice while still working with probabilistic models.

For this project, I will be looking mostly into the theoretical convergence and somewhat with implementation limitations. The theoretical convergence issue I will be looking into is the relationship between neural network size and similarity to the target function. For simplicity, I believe this will be how quickly a Bayesian network can approximate a gaussian process. As for implementation limitations, this will be what we can feasibly calculate and what tricks/approximations must be done to complete the task. Some examples of this could be experimenting with different priors, limiting Monte Carlo sampling size, or the reparameterization trick. To this end, another, smaller aspect of the project will be what approximations are worth doing with respect to the theoretical convergence, training speed, and accuracy.

Past Work:

The theory on this topic is very well established and there are many papers that empirically explore this connection with different approaches. The difficulty therein lies in whether I can explore a new route, perhaps looking at variations of recent papers, or if I am simply going to replicate (or closely replicate) past experiments with heavier ties to theoretical analysis.

A good foundational reference for this topic is: https://link.springer.com/chapter/10.1007/978-1-4612-0745-0_2, Bayesian Learning for Neural Networks, specifically Chapter 2: Priors for Infinite Networks. Many more recent papers on the topic cite the theory established in this book. Something very similar to what I propose is the much more recent paper: https://arxiv.org/pdf/2006.10541.pdf, Exact Posterior Distributions of Wide Bayesian Neural Networks. This paper builds on another from 2018 to better explore the theory in an empirical way.

Approach:

There are several possible implementation approaches possible. There are many github repositories available that have implementations of Bayesian Neural Networks that I would be able to use (and possibly tinker with), and the Gaussian Process has implementations in several machine learning libraries. It is also possible to try my own implementation, but I am unsure of how flexible it would be. This approach may simply be me spending most of my time trying to optimize code to run in parallel and utilize the GPU rather than focusing on the literature and experimentation. (This would be the only way it ties in with my research with Dr. Lowenthal).

My project would have to start with and be motivated by the established theory. First, I would like to look more into the literature and math behind it in the hopes of seeing how tight the theoretical bound is and what can be manipulated. Specifically, how rapidly does the network converge, with what probability, with what error bound, and does it depend on the complexity of the GP. Of course, there may not be an answer to each of these questions, but each one can still be experimented on and empirically evaluated.

I brought up four aspects of the research I wanted to investigate: the speed of the convergence, if the convergence has a probabilistic nature, if the convergence has an established error bound, and does any of this change with the complexity of the Gaussian Process. I believe this naturally leads to experimental variables. I will conveniently label the aspects as “speed”, “probability”, “error”, and “complexity.”

Speed: This will be the main focus, where the size of the neural network is increased, and the rate of this increase is compared to the rate at which the similarity grows, for some definition of similarity.

Probability: This directly corresponds to simply having multiple trials, perhaps training on all possible data subsets (or a wide variety if computationally annoying), to see if even with a very large neural network, you could still get a bad model with unlucky data.

Error: This is like probability, except we must directly compare the outputs of the true Gaussian Process to our model, and we must see if there is a bound on the distance to the outputs, for some definition of distance. This could be difficult to evaluate due to there being both sample size and model size variability.

Complexity: This corresponds to running the entirety of the experiment for different gaussian processes of varying complexity. (Whatever complexity may mean for a gaussian process). This would be particularly interesting if there were no relationship between complexity and convergence, or if there is a very clear relationship between complexity and convergence.

Evaluation Methodology:

This is heavily dependent on the theoretical answers. I could feasibly run all these experiments without looking at the theory, but I may not know what my results mean and how to evaluate them without looking at the theory. Further, the theory could provide insight into better ways of isolating certain variables. So as of now, I am not certain I can do a proper evaluation without fully understanding the theory.

With theory, it would be as simple as seeing if the empirical results hold within the bounds. More interestingly, I could also explore whether the bounds are empirically tight or if there appears to be some gap in the bounds. A gap could indicate that we should find a tighter bound on the theory side, potentially changing our understanding of the model.

I could also use other papers' results as a metric of evaluation. As stated previously, many other papers have very similar experiments, and I could simply compare my results to theirs and see if there is agreement or not.

Comments:

The experimental side sounds interesting enough, and I have expanded much more here than with my research proposal for CSc 588. I just need to know how pI can make it clear that there are two distinct parts of this project, one for CSc 588, and one for CSc 696.