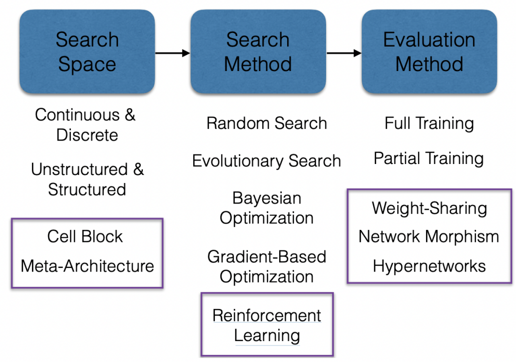
Graduate Research Plan: Graph Generation for Neural Architecture Search

Neural architecture search (NAS), a hyperparameter optimization problem, automates the design of an optimal neural architecture for a specific task or dataset, eliminating or reducing the need for expert-designed networks for a given problem set. NAS workflows generally consist of three components: search space, search method, and evaluation method, as seen in Figure 1.



Despite widespread interest in NAS, evaluating the success of specialized NAS frameworks relative to more widespread hyperparameter optimization methods is difficult because those more common methods have not yet been tested on NAS benchmarks [3]. This lack of crossover testing makes understanding the value of specialized frameworks almost impossible.

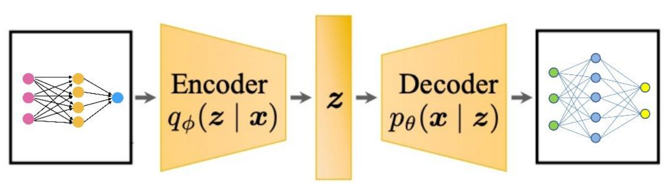
Simultaneously, there is a growing body of research on the generation of realistic graphs that meet certain desired criteria in a variety of fields, from protein design to program synthesis [9]. However, generative graph neural networks have not yet been applied to neural architecture, despite the ease of representing a neural network as a directed acyclic graph.

I propose (1) to develop an entirely new search method relying on generative graph models and (2) to test the effectiveness of this method against previous work in the NAS space and against more common hyperparameter optimization techniques. This proposed research exists at the intersection of a number of unanswered questions both in neural architecture search and in generative graph neural networks.

Proposal

Part I: Search Method

In this research, I will apply a latent-variable graph generation approach to neural architectures in order to generate an optimal neural architecture for a specified dataset. The approach can then be broadened to other methods of graph generation, such as graph structure learning or graph recurrent neural networks [9,10].



The latent variable approach consists of an encoder, sampler, and decoder as seen in Figure 2. The encoder represents each graph as dense, continuous vectors and outputs the parameters of a stochastic distribution, which can then be sampled [9]. The decoder uses the sampled latent representations to generate new graph architectures. I will use a variational autoencoder (VAE) as the encoding function, which estimates the distribution of the graphs by maximizing the Evidence Lower Bound [2]. I will then randomly sample the distribution before feeding the latent representations to the decoder, which will reconstruct the graph representations sequentially as proposed in “Constrained Graph Variational Autoencoders for Molecule Design” [5].

The first inputs to the variational autoencoder will be generated randomly from the search space. Each neural architecture will undergo partial training and then be evaluated against a held-out test set. The best models of each generation, as defined by validation accuracy, will then be encoded and sampled. In this way, new neural architectures can be generated and tested. The decoder can be configured to confine generated graphs to the original search space by inputting constraints on the graph structure [5].

Part II: Evaluation

This methodology will be tested against the NAS benchmarks available through NAS-Bench-Suite [6]. I will consider the same search space and evaluation method as DARTS in order to enable direct comparison to existing work [4]. To quantify the benefit of specialized NAS techniques over more popular and better-tested optimization methodologies, I will also evaluate the results of standard hyperparameter optimization methods, from ideas as simple as momentum to state-of-the-art Bayesian optimization accelerators, on the NAS-Bench-Suite [6]. Testing these models will require significant computational resources, for which I plan to utilize JetStream.

Intellectual Merit

This work will evaluate the performance of NAS-specific optimization methods in comparison to general hyperparameter optimization methods, while studying the intersection of two expanding fields: NAS and generative graph models. The primary result of this research will be a novel search method based on generative graph models that is also search space agnostic. Secondary results will include thorough evaluation of the performance of the new search method relative to both widely-implemented NAS methodologies as well as hyperparameter optimization methodologies that are not specialized for the neural architecture search problem. These results will contextualize successes in NAS relative to simpler and less time-consuming approaches.

Broader Impacts

Improving neural architecture search would dramatically increase the accessibility of machine learning tools to those without substantial background knowledge. Most deployed neural architectures are hand-designed by experts on an ad-hoc basis, which creates a large bottleneck in the production of high-quality architectures for specific problem sets. Machine learning experts are not readily available in many parts of the world, where local populations lose out on the potential benefits of machine learning. Smoothing the process of neural architecture design could enable wider access to next-generation technologies across the globe. Additionally, scientists and researchers across many disciplines stand to benefit from workflows improved and augmented by machine learning without the intervention of a resident computer scientist.

References:

[1] Gao Y, Yang H, Zhang P, Zhou C, Hu Y (2019) GraphNAS: Graph Neural Architecture Search with Reinforcement Learning. http://arxiv.org/abs/1904.09981
[2] Kingma DP, Welling M (2014) Auto-Encoding Variational Bayes. http://arxiv.org/abs/1312.6114
[3] Li L, Talwalkar A (2019) Random Search and Reproducibility for Neural Architecture Search. http://arxiv.org/abs/1902.07638
[4] Liu H, Simonyan K, Yang Y (2019) DARTS: Differentiable Architecture Search. http://arxiv.org/abs/1806.09055
[5] Liu Q, Allamanis M, Brockschmidt M, Gaunt AL (2019) Constrained Graph Variational Autoencoders for Molecule Design. http://arxiv.org/abs/1805.09076
[6] Mehta Y, White C, Zela A, Krishnakumar A, Zabergja G, Moradian S, Safari M, Yu K, Hutter F (2022) NAS-Bench-Suite: NAS Evaluation is (Now) Surprisingly Easy. http://arxiv.org/abs/2201.13396
[7] Yang C, Xiao Y, Zhang Y, Sun Y, Han J (2020) Heterogeneous Network Representation Learning: A Unified Framework with Survey and Benchmark. http://arxiv.org/abs/2004.00216
[8] You J, Ying R, Ren X, Hamilton WL, Leskovec J (2018) GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. http://arxiv.org/abs/1802.08773
[9] Zhu Y, Du Y, Wang Y, Xu Y, Zhang J, Liu Q, Wu S (2022) A Survey on Deep Graph Generation: Methods and Applications. http://arxiv.org/abs/2203.06714
[10] Zhu Y, Xu W, Zhang J, Du Y, Zhang J, Liu Q, Yang C, Wu S (2022) A Survey on Graph Structure Learning: Progress and Opportunities. http://arxiv.org/abs/2103.03036