Automatic Hyper-parameter Tuning with Genetic Algorithms and Transfer Learning

Presenter: Eashan Singh, UCF CS/Math Undergrad

Motivation

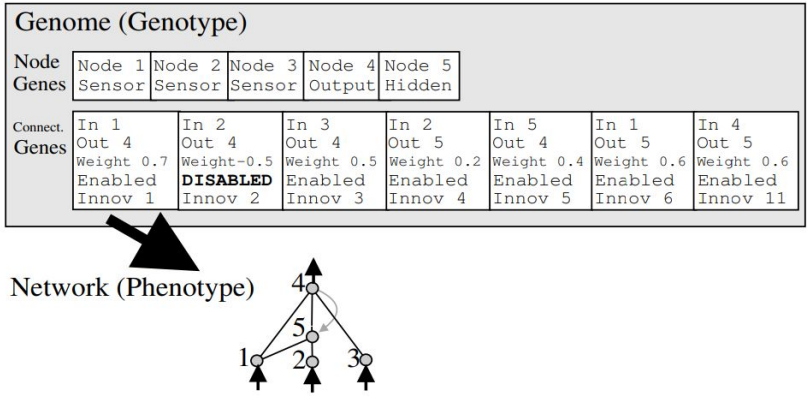
1) Deep learning models are becoming increasingly intricate with unique topologies and improved model architectures. Fine-tuning hyper-parameters can become a very **time intensive** task and difficult to perform manually.
2) Research has shown **genetic algorithms**(GA) offer an alternative way to find optimality of parameters and for exploration of the hyperparameter search space.
3) Leveraging pre-trained models via **transfer learning** can potentially help improve model accuracy and prove an alternative for other resource-intensive deep learning techniques, making it more viable in a clinical environment.
4) This project also aims to improve the models generalizability via **data augmentation** techniques to increase the diversity and size of the dataset, with the goal is to improve model performance on unseen data.

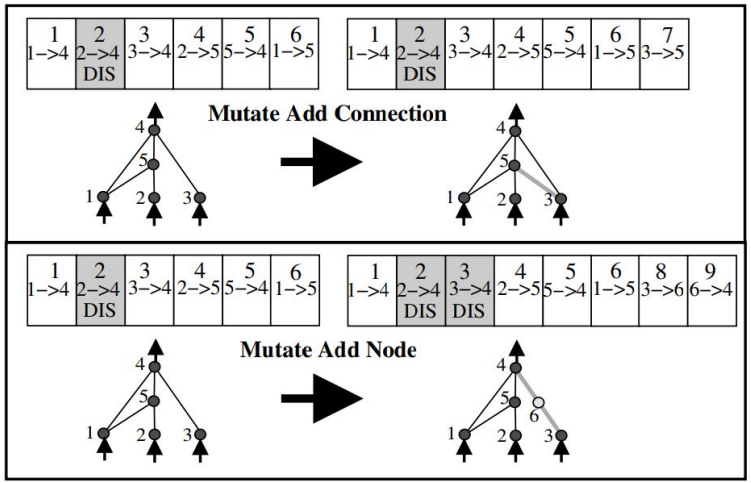
Problem Statement

- This research aims to develop a deep learning model for brain tumor classification that combines transfer learning and genetic algorithms(GAs) for optimized hyper-parameter tuning.
  - Using the BraTS2020 Dataset
  - With a pre-trained CNN network like ResNet
- Hyper-parameters this paper will study:
  - Learning Rate
  - Batch Size
  - Number of Layers

Related Work

- Stanley and Miikkulainen introduced NeuroEvolution of Augmenting Topologies(NEAT) where they utilized a Genetic Algorithms for evolved network topology and connection weights for solving Reinforcement Learning problems. [1]
- The top image represents a genome to phenotype mapping example.
- The bottom image shows 2 examples of structural mutations: adding a connection(weight) and adding a node.

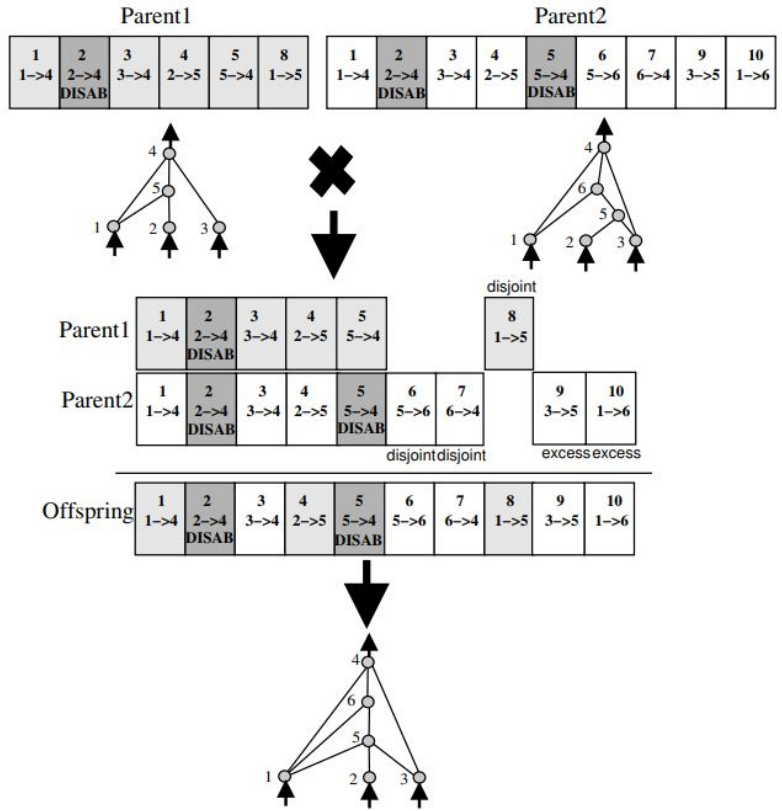




[1]

Related Work contd.

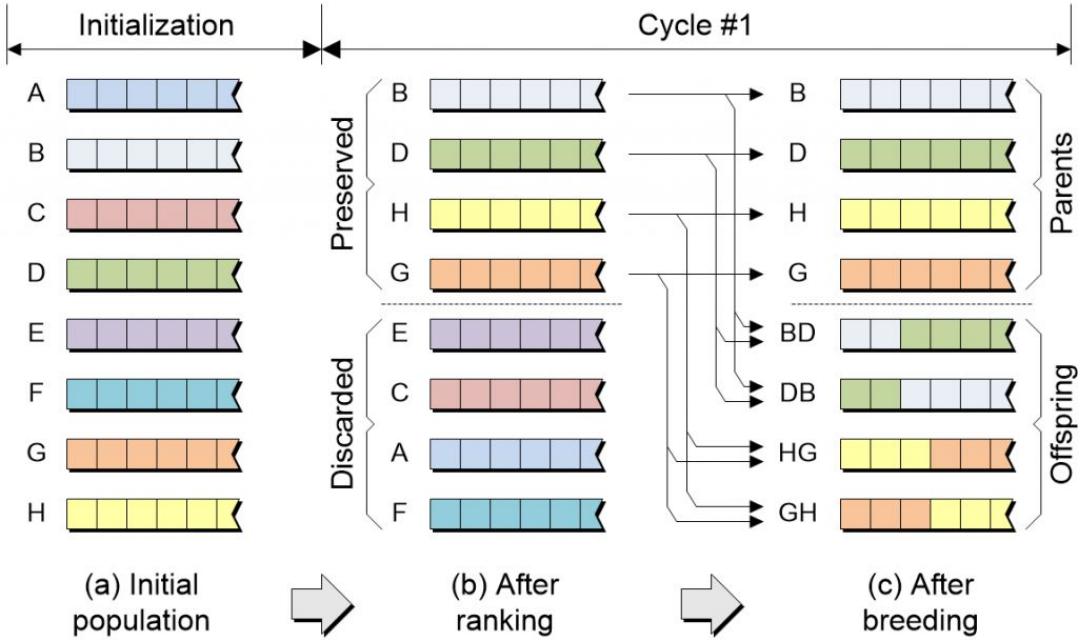
- NEAT exploited the properties of structure and history for incremental growth of network structures.
- The emphasis of this research was to show that genomes could grow in complexity as necessary and that optimizations of minimal structures helped to evolve networks with increasingly complex topologies.



[1]

Background: Genetic Algorithms

- **Genetic Algorithms** are a class of optimization algorithms which take inspiration from **natural selection** and survival of the fittest strategy.
- An initial population is made based off an encoding of candidate solutions and an **elitism** structure is used to select parent candidates for reproduction.
- Parents undergo **crossover** and **mutation** and child offspring become the parents of the next generation
- The algorithm runs until an optimal solution or stopping criteria is met.



[2]

Methodology

- A binary bit encoding will be made to represent the the hyper-parameters of the CNN model(chromosome candidate solution of the GA)

  *Encoding **E**<learning rate, batch size, # of layers>*

- Initialize a population with **N** encodings **E**
- For **R** runs:
  - For **G** generations:
    - Evaluate the fitness of all candidate solutions(evaluating hyperparameters on the classification task)
    - Employ a **tournament selection** strategy to pick parents for reproduction
    - Perform **crossover**
    - Perform **mutation**
    - Offspring of generation **g** become population of generation **g+1**
- Run until stopping criterion is met or **R** runs.

Experimental Evaluation

Genetic Algorithm Tuning. In an attempt to optimize the parameters of our CNN, experimentation is required to find the ideal hyper-parameters of our genetic algorithm (GA). Testing will be required to find the following:

1. Population size
2. Mutation type (uniform random, gaussian, polynomial)
3. Mutation rates (0, 0.005, 0.01, 0.05)
4. Crossover type (single-point, two-point, blend, simplex, linear, simulated binary)
5. Crossover rates (0.01, 0.05, 0.1, 0.5, 0.9)
6. Selection strategy (tournament, rank, proportional)

Progress

1) [COMPLETED] Implement the generic SGA(standard genetic algorithm) infrastructure needed to run the experiment. Evaluated on:
   a) Traveling Salesman Problem
   b) Point Scattering Problem
2) [COMPLETED] Evaluate a baseline model(ResNet-18) on the given classification task. Tested also with basic data augmentations.
3) [IN-PROGRESS] Integrating the project specific encoding(slide 7.) with the SGA
4) [IN-PROGRESS] Running experiments with varied SGA hyper-parameters
5) [TO-DO] Re-run experiments for better model generalizability (data augmentations)
6) [TO-DO] Compare performance of the baseline classifier to the optimized hyper-parameters and draw conclusions.
7) Stretch Goal*: Compare the performance of hyper-parameter optimizations made by the SGA to other optimization techniques like Particle Swarm Optimization, Teaching Learning Based Optimization, and Differential Evolution.

Potential Roadblocks

- Computational overhead challenges
  - Number of functional evaluations = No. runs * No. generations * Size of population* No. epochs * No. evals per epoch
  - Quality of solutions very much dependent on population size
  - Larger models will require more computational overhead. May need to evaluate on smaller model architectures

References

[1] Stanley, Kenneth O., and Risto Miikkulainen. "Efficient reinforcement learning through evolving neural network topologies." *Proceedings of the 4th Annual Conference on genetic and evolutionary computation*. 2002.

[2]Maxfield, M. (2020, July 9). When genetic algorithms meet Artificial Intelligence. EEJournal. Retrieved April 18, 2023, from https://www.eejournal.com/article/when-genetic-algorithms-meet-artificial-intelligence/