Note: This is the introduction to my research paper. I am still in the process of writing the entire paper and incorporating my results as I have recently finished my data analysis. The paper is required in the application form, so I am submitting what I have so far.

Training Neural Networks with a Genetic Algorithm for Obstacle Avoidance in Simulated Autonomous Drones

Will Baumgartner

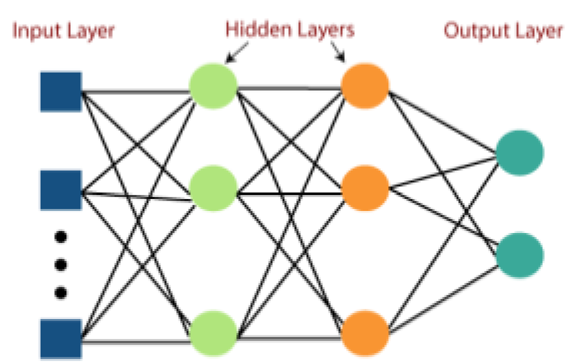
1. Introduction

The use of unmanned aerial vehicles (UAVs; or drones) for a wide variety of applications has become increasingly popular over the last few years. Along with this, machine learning has become a more viable solution for controlling autonomous machines.

One application where autonomous UAVs have been proved viable is afforestation and reforestation. UAVs are viable for this task due to their speed, scalability, cost effectiveness, and safety. UAVs can also be remotely controlled, take autonomous flying paths around obstacles using aerial and satellite data, have more mobility and precision, and have cheap maintenance and usage costs (Mohan et al., 2021). In addition, UAVs could also be used for package delivery. UAVs can provide on-demand, inexpensive, and convenient access to nearby items, like medicine and groceries. UAVs are able to minimize cost and delivery time compared to their ground based alternatives. Their paths could even be optimized further, providing uniform network or surveillance coverage along the way to their destinations (Khosravi et al., 2019).

To control the drones in this study, we utilize neural networks, specifically multilayer perceptrons (MLPs). MLPs take a set of input

Figure 1



Multilayer Perceptron (javatpoint.com)

values and transform them into a set of output values, utilizing hidden layers with neurons as seen in Figure 1. Each node is connected to all of the nodes in the previous layer of the network, making MLPs fully connected neural networks. Each connection corresponds to a weight, and we sum the product of the previous nodes and their weights to get the value of the current node. Each node also has a bias, which is added to the product sum. We can easily model this using matrices and matrix products, multiplying the input vector by matrices of weights until we reach the output layer. The input values move forward through the model without looping back (recursion), so this is a feedforward neural network. Initially, the weights and biases are random, yielding nonsensical output for any given input. However, we can optimize these weights and biases to produce the output we are looking for, which is called training. There are many popular optimization methods, but in this study, we utilize a genetic algorithm. It has been shown that the use of a genetic algorithm can provide better results in some cases for training a feedforward neural network than the traditional techniques of backpropagation (Gupta and Sexton, 1999).

A genetic algorithm is traditionally used to model natural selection within a simulated population. Here, we use a genetic algorithm to select for neural networks that control drones well, and to remove drones that are useless. To start, we initialize the neural networks with random weights and biases, each drone receiving one neural network controller. Some drones will perform better than others, but initially most are incapable. We score the drones' fitness based on certain criteria (Do the drones hit anything? Do they make it to their target?), and eliminate a certain percentage of them that scored low. After this, we crossbreed the remaining successful drones to repopulate, combining weights and biases from randomly chosen parents. After we have a new population, we can perform simple mutation (randomizing a few of the weights and biases) on the new drones in order to get some variation. After running the simulation many times, the drones will slowly increase in capability, with the less fit drones dying off and the more fit drones producing even more fit children. Essentially, we are optimizing the neural networks to produce a drone that best fits the given criteria. The only control over the training we have is the fitness criteria, as well as various hyperparameters for the genetic algorithm. These hyperparameters dictate things like how many drones we eliminate each generation, how fast the neural networks mutate, and how many drones are in the population.

In this study, we use a genetic algorithm to train autonomous drones equipped with neural network controllers in an attempt to optimize speed and obstacle avoidance. It's easy enough to get a drone to stay up in the air, so this study focuses on the more complex task of making sure that there are no costly collisions between drones or with obstacles, and that the drones make it to their waypoints in a respectable amount of time. One study has similarly attempted to automate drone pathing to avoid obstacles by processing camera input with convolutional neural networks, which was very successful (Amer et al., 2017). Here, we also attempt to perform obstacle avoidance, but with positional information and intercommunication between drones rather than the limited information acquired from a camera. Another study has also utilized genetic algorithms, tuning PID controllers for drone flight and showing that genetic algorithms are more than viable for complex tasks like obstacle avoidance in drones (Elajrami et al., 2021). Another study similar to this one has employed a genetic algorithm to optimize the path of drones to provide the most amount of area coverage while using the least amount of energy. The optimized algorithm consumed 2-5 times less energy than that of a traditional greedy algorithm by reducing the number of turns while covering all the waypoints (Shivgan and Dong, 2017). Again, our study aims to optimize the path of autonomous drones using a genetic algorithm by minimizing collisions as well as travel time. It is hypothesized that the optimized drone control algorithm will result in a slightly slower time than the straight path algorithm, but result in much safer travel with extremely few collisions.

Works Cited:

Amer, K., Samy, M., Shaker, M., & Elhelw, M. (2021). Deep convolutional neural network based autonomous drone navigation. *Thirteenth International Conference on Machine Vision*. https://doi.org/10.1117/12.2587105

Elajrami, M., Satla, Z., & Bendine, K. (2021). Ajr Drone Control using the Coupling of the PID Controller and Genetic Algorithm. *Communications - Scientific Letters of the University of Zilina*, *23*(3), C75–C82. https://doi.org/10.26552/com.c.2021.3.c75-c82

Gupta, J. N., & Sexton, R. S. (1999). Comparing backpropagation with a genetic algorithm for neural network training. *Omega*, *27*(6), 679–684. https://doi.org/10.1016/s0305-0483(99)00027-4

Khosravi, M., Enayati, S., Saeedi, H., & Pishro-Nik, H. (2021). Multi-Purpose Drones for Coverage and Transport Applications. *IEEE Transactions on Wireless Communications*, *20*(6), 3974–3987. https://doi.org/10.1109/twc.2021.3054748

Mohan, M., Richardson, G., Gopan, G., Aghai, M. M., Bajaj, S., Galgamuwa, G. A. P., Vastaranta, M., Arachchige, P. S. P., Amorós, L., Corte, A. P. D., de-Miguel, S., Leite, R. V., Kganyago, M., Broadbent, E. N., Doaemo, W., Shorab, M. A. B., & Cardil, A. (2021). UAV-Supported Forest Regeneration: Current Trends, Challenges and Implications. *Remote Sensing*, *13*(13), 2596. https://doi.org/10.3390/rs13132596