Dawson Byrd, Sanath Govindarajan, Thomas Wang
Asymptotic Analysis, CSC505 Design Project

N = number of obstacles, M = number of neural networks in given simulation

Neuron:

getWeights():

For a given neuron A, getWeights() method loops through all edges connecting A. This runs in O(k) time, where k is the number of nodes in the layer below A. The worst-case scenario for this is when A is adjacent is the output layer (where the previous layer has constant 20 nodes), hence the runtime is O(1).

NeuralNet:

Notes: The number of nodes in the input layer is O(1). The number of nodes in the following layers are all O(1). Hence the number of edges in the neural network = O(1)*O(1) + k*O(1) = O(1).

getWeights(): This method runs through each of the weights in the neural network, and hence takes O(1) time

getOutput(): getOutput() runs through each of the layers (from bottom to top) in the neural network activates each of its nodes. That is, it receives outputs from node in previous layer, and produces an output value according to the given neuron's edge weights and activation function). Hence, asymptotically this is equivalent to looping through each weight, which runs in O(1) time.

setWeights(): setWeights() ~ getWeights(), hence this method also runs in O(1) time.

config(): The configuration method is composed of each of the functions above, all of which have runtimes O(1), hence the runtime of config() is O(1).

Sim:

checkHit() loops through each obstacle, and hence runs in O(n) time

update(): Update is composed of the following known functions: checkHit(), setInputs() agetOutput(). Since checkHit's runtime has the greatest order of magnitude, the runtime of update is O(N).

Layer:

All methods in the Layer class have O(1) runtime, since the worst case input size is constant.

Generation:

FPS(): loops through all of the networks in the member array, hence it is O(M)

reproduction():

> @asexual Runs through the top k=M/10 neural networks and assigns them to the first k position in the children array. Hence it runs in O(M) time.
>
> @crossover Contains a double loop that visits each unordered pair of NeuralNets in selection (which has length N/5). There is also a reference to NeuralNet.config() and NeuralNet.setWeights(), all of which run in constant time. Hence the runtime for the crossover operation is O(M^2).
>
> @mutation The mutation operation simply loops through each neural network in the **children** array and mutates them by a small factor. Hence the runtime is O(M).

Since reproduction is composed of these three functions, its runtime ~ the most computationally expensive. Hence reproduction runs in O(M^2) time.