

# Genetic Algorithms with Neural Networks Generic Artificial Player

Elias Obeid and Martin Madsen and Kent Caspersen<sup>1</sup>

## 1 Preliminaries

### 1.1 Genetic algorithms

A *genetic algorithm* belongs to the class of evolutionary algorithms. This class of algorithms is inspired by natural evolution as seen in Biology. Genetic algorithms are search algorithms which imitate the process of natural selection.

**Populations** A genetic algorithm has a *population* of *individuals*. These individuals can be initialised randomly with a fixed population size. The goal for these individuals is to solve a problem optimally. Each individual in the population has a fitness level that defines the individual's ability to solve a given problem.

**Genes** Each individual has specific *genes*. Genes constitute the DNA of the individual. These genes are an encoding of some attribute or skill the individual has. The encoding of genes is up to the creator, but it is most often encoded as bitstrings.

**Crossovers and mutations** In natural evolution, a pair of individuals come together to produce a new individual, a child individual, with genes from both of the parent individuals. The process of breeding is done by performing a *crossover* of the two parent individuals' genes. A crossover point is defined and one part of each of the parents is copied and combined to form a new set of genes for the child individual.

*Mutations* can occur randomly at any point in time. While the genes are encoded as bitstrings, then a mutation arbitrarily changes one of the bits. This ensures that the population can evolve if no progress would be made if the genes did not allow it.

**Fitness function** A *fitness function* must be defined to calculate the fitness level for each individual. This function is used to define the most fit individuals. The higher the fitness level is of a given individual, the higher the chance it has to be chosen to reproduce with another individual.

### 1.2 Neural network

The artificial *neural network* is a computational model inspired by the brain. Neural networks are constructed of connected computational units called *neurons*.

**Neurons** We distinguish between *input*, *hidden*, and *output neurons*. Input neurons provide and forward the input for the network. Hidden neurons are the units between the input and the output neurons. Based on the values of the output neurons an action is to be performed. Neurons receive one or more inputs and produce a single output. The output is forwarded to the next layer of neurons in the network.

The network consists of layers of neurons. All these neurons in one layer are connected to all the neurons of the next layer. Each connection has an associated weight which is used to produce the single output.

**Transfer function** Each computational unit uses a transfer function to produce an output. In this article we use the sigmoid function:

$$S(t) = \frac{1}{1 + e^{-t}}$$

The sigmoid function produces a value between 0 and 1. The output,  $y_l$ , of the  $k$ th neuron with  $m$  inputs to the given neuron, is calculated as:

$$y_l = \varphi \left( \sum_{k=1}^m w_{lk} x_k \right)$$

where  $\varphi$  is the transfer function,  $x_k$  is the value of input  $k$ , and  $w_{lp}$  is the weight of the connection between neuron  $l$  and  $k$ .

---

<sup>1</sup> Aalborg University, Denmark, email: d601f14@cs.aau.dk