KTH Computer Science
and Communication

Title

Subtitle

BJÖRN TEGELUND
JOHAN WIKSTRÖM

Bachelor's Thesis at CSC
Supervisor: Petter Ögren
Examiner: Mårten Björkman

TRITA xxx yyyy-nn

Abstract

This is a skeleton for KTH theses. More documentation regarding the KTH thesis class file can be found in the package documentation.

Referat

Titel på svenska

Denna fil gfwfqwfwqfqer ett avhandlingsskelett. Mer information om LaTeX-mallen finns i dokumentationen till paketet.

Contents

Chapter 1

Introduction

1.1 Background

Genetic algorithms are algorithms that emulate evolution to achieve a optimal solution to a problem[1]. These algorithms have many uses and we wanted to investigate whether the phenomena which occur in nature due to evolution also occurs when using genetic algorithms. In nature evolution has spawned a wide variety of survival strategies such as adaptation, camouflage colours and mimicry. Genetic algorithms however, are very simplified mathematical models of these mechanisms which means that these phenomena might not occur at all.

1.2 Scope and Objectives

Main objective:
Compare RBF-based and "linear" brains.
By: Forcing them to adapt certain genetic phenomena such as adaptation (learning), co-evolution and extinction, mimicry, group forming and co-operation prioritised in the order presented (as increasingly complex behaviour and difficulty to simulate).

Questions:
How quickly do they learn?
Can both kinds of brains do the same things?
Pros and cons for each brain architecture
RBF-based predators vs linear prey, vice versa.

Evolution of brains using RBF functions and evolutionary algorithms.

1.3 Achievements

[1] Proof

Chapter 2

Technical Overview

2.1 Genetic Algorithms

2.1.1 Overview

What are genetic algorithms?
How are they typically used?
How do they relate to genetics and evolution?
Overview of algorithm

2.1.2 Selection

2.1.3 Fitness

2.1.4 Mutation

2.1.5 Crossover

2.2 Radial Basis Functions

2.2.1 Overview

A radial basis function (RBF) is a bell-shaped function which value depends on the distance from some origin, denoted μ in the formula. Radial basis functions are commonly used in neural networks as a way to encode input information. They are favourable to use as they have locality, something which linear functions do not. In particular they are used for function approximation, as any function can be approximated as the sum of a number of weighted radial basis functions. A property of radial basis functions which can both be interpreted as an advantage and disadvantage is that their value never exceeds 1, compared to a linear function which can grow to infinitely high or low numbers.

Radial basis functions are commonly implemented using a formula such as 2.2, which is a three-dimensional function centred around (μ_x, μ_y, μ_z). The width of

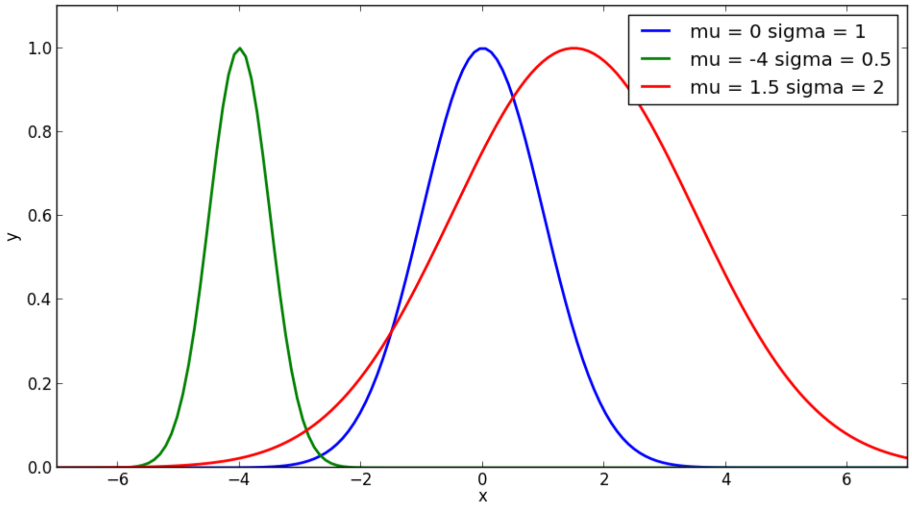

Figure 2.1. Three one-dimensional RBFs with varying μ and σ values.

$$f(x, y, z) = \exp(-(\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2} + \frac{(z-\mu_z)^2}{2\sigma_z^2})) \tag{2.1}$$

Figure 2.2. A three-dimensional radial basis function.

the bell-curve in each dimension is determined by σ_x, σ_y and σ_z respectively.

Chapter 3

Implementation

3.1 Model

3.1.1 Simulation in Python

Square world with walls, circular inhabitants with antennae
Good and bad food sources
Predators?
Possible modifications to enforce different behaviour?
What libraries will we use? Why?
Optimisations?

3.1.2 Methods of Enforcing Behaviour

Adding input
Additional terms in brain calculations
Adding predators
More kinds of food
Being able to see more things in the world
Placing objects into the world

3.1.3 RBF-Based Decision Making

When input is received by a creature, it is in form of eight numbers, four for each antenna. Three of the inputs for each antenna are the red-, green- and blue components of the currently detected object's colour. The fourth input is zero when no object is detected and one when an object is. The three colour-based inputs are then used in several three-dimensional radial basis functions. For each antenna a Δr (change in rotation) and Δs (change in speed) is computed by summing the function values and normalising them. The Δr and Δs for each antenna are then

$$\Delta s_{left} = f(\mathbf{x}_{left}) \tag{3.1}$$

Figure 3.1. Calculating the Δs for the left eye, using the input vector $\mathbf{x}_{left}$ corresponding to the colours of the object detected using the left antenna.

grouped and used by the creature to change it's rotation and speed. See equation 3.1.

Each radial basis function also has a σ and μ which are decided by the creature's genes. σ and μ are in the ranges of $[0, 1]$ and $[-1, 1]$ respectively. In practice this means that the creatures' change in speed and rotation, when detecting an object in each antenna, depend on their genetic makeup.

The difference between using radial basis functions and linear functions is that you have a much larger possibility to approximate any decision-making strategy. For example an RBF-based brain could make the distinction between different shades of green and thus react differently to them, while a linear function could only decide if more green is better or worse.

3.1.4 Genetic Algorithm

What kind of genetic algorithm do we plan to use?
Fitness, crossover, mutation, selection, which ones?
Flow chart of our algorithm
NSGA-II

3.1.5 Experiments

(In every experiment linear and RBF are compared, eventually mixed)
Interested in:
Time to maximal fitness
Highest possible fitness

- Adaptation 1. Finding and eating food
- Adaptation 2. Finding and eating food, avoiding "bad" food
- Co-evolution and extinction 1. Predator vs prey, prey eats food as in first experiment. (bushes are bad for predators)
- Speciation and co-operation 1. Allow the creatures to evolve which "colour" food they can eat, attempt to create two separate species from one, one species eating green bushes and one red.
- Mimicry 1. Making the prey mimic bushes or predators to avoid being eaten.

- Mimicry 2. Predators attempt to mimic bushes to make prey run into them.
- Group forming 1. Rerun all previous experiments, but with the ability to detect the nearest creature of the same species.

Chapter 4

Results

4.1 Simulation results

4.1.1 Observed Behaviours

4.2 Discussion

4.2.1 Constraints and problems

Performance problems
Other unexpected problems?

4.3 Conclusions and Future Work

References

1. Montana, D. J., and Davis, L. (1989, August). Training feedforward neural networks using genetic algorithms. In *Proceedings of the eleventh international joint conference on artificial Intelligence* (Vol. 1, pp. 762-767).

2. Langton, C. G., and Shimohara, T. (Eds.). (1997). *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems* (Vol. 5). Mit Press.

Appendix A

RDF

And here is a figure

Figure A.1. Several statements describing the same resource.

that we refer to here: A.1