THE UNIVERSITY of York

Cartesian Genetic Programming encoded Artificial Neural Networks: A Comparison using Three Benchmarks

Andy Turner & Julian Miller

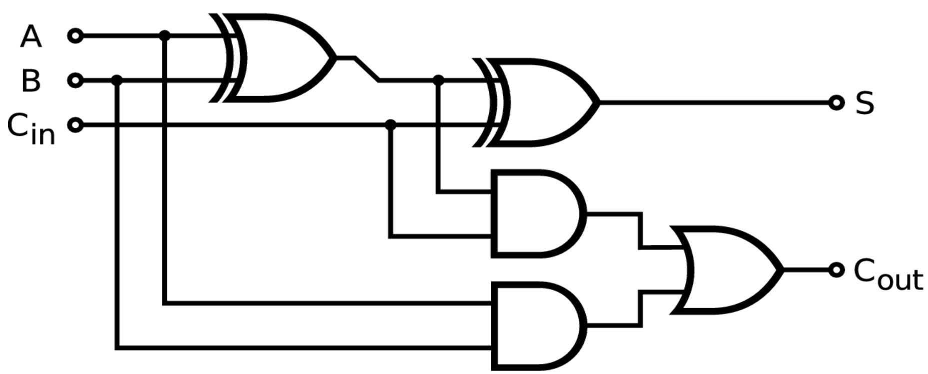
at568@york.ac.uk & Julian.miller@york.ac.uk

Cartesian Genetic Programming

First Implemented by J. F. Miller & P. Thomson, 2000
Web page: http://www.cartesiangp.co.uk/

Key Points

- Form of Genetic Programming
- Cyclic and acyclic graphs
- Any data type e.g. ints, floats, images, videos ...
- Any function e.g. XOR, sigmoidal, sin ...
- Inbuilt neutrality and genetic drift (Miller et al, 2006)
- Natural resilience to bloat (Miller, 2001)
- Typically uses a (1+4)-ES
- Mutation only (no crossover)
- Not just for circuits



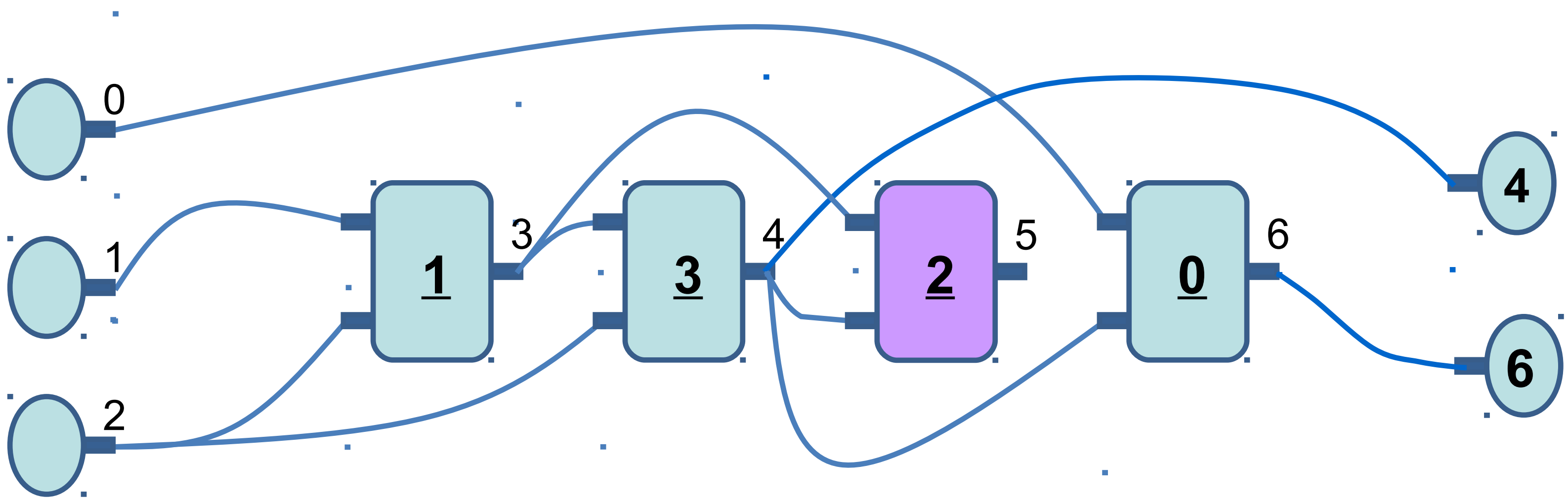
CGP Structure

Genotype

1 1 2	3 3 2	2 3 4	0 0 4	4 6

One row, four columns

Phenotype



Andy Turner
at568@york.ac.uk

CGP and Neural Networks

First published by Maryam. M. Khan et al, 2010

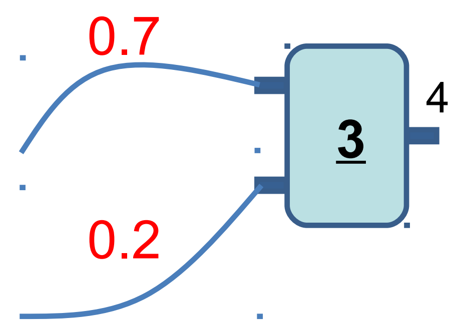
Minor changes to CGP to encode Neural Networks

- F — Function i.e. sigmoid, radial basis...
- C — Unchanged
- W — Connection Weight
- Inputs — Unchanged
- Outputs — Unchanged

3 3 2 0.7 0.2

CGPANN Features

- Evolves Weights
- Evolves Number of Neuron
- Evolves Topology
- Evolves Arity of Neurons (indirectly)
- Evolves Functions
- All of the advantages of CGP



Andy Turner
at568@york.ac.uk

NeuroEvolution

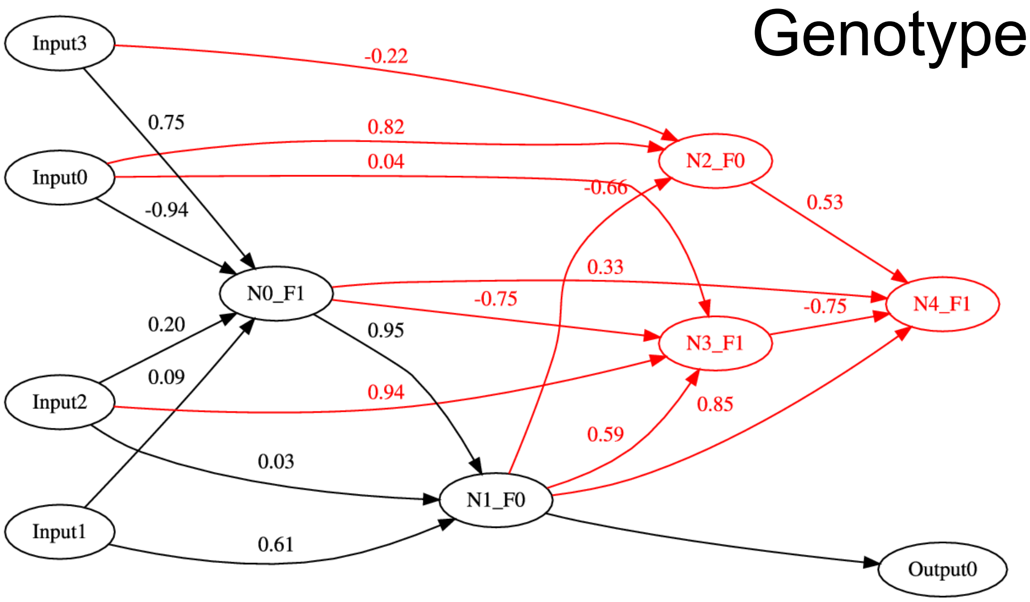
Weight Evolution
- Does not require differentiable neuron functions
- Does not require a precise fitness function
- Does not struggle to train deep topologies
- Searches weight space

Topology Evolution
- Does not require a suitable topology to be known in advance
- Produces topologies which would not usually be considered
- Searches topology space

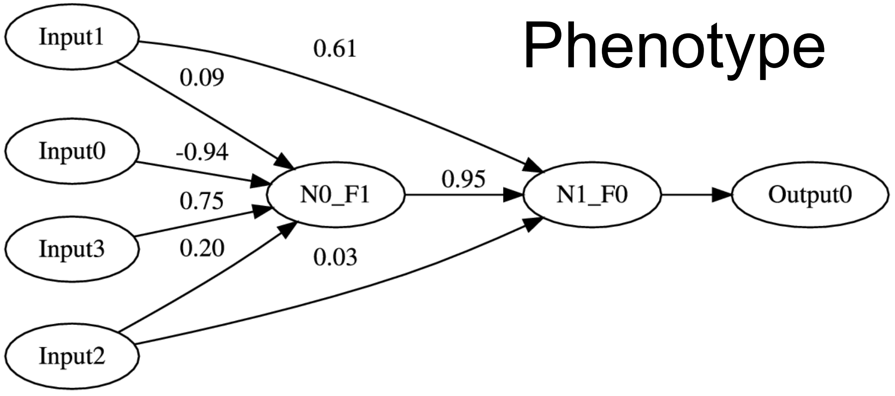
Node Transfer Evolution
- Does not require suitable node functions to be known in advance
- Can easily use a mix of node functions
- Searches function space

Andy Turner at568@york.ac.uk

Inactive Nodes

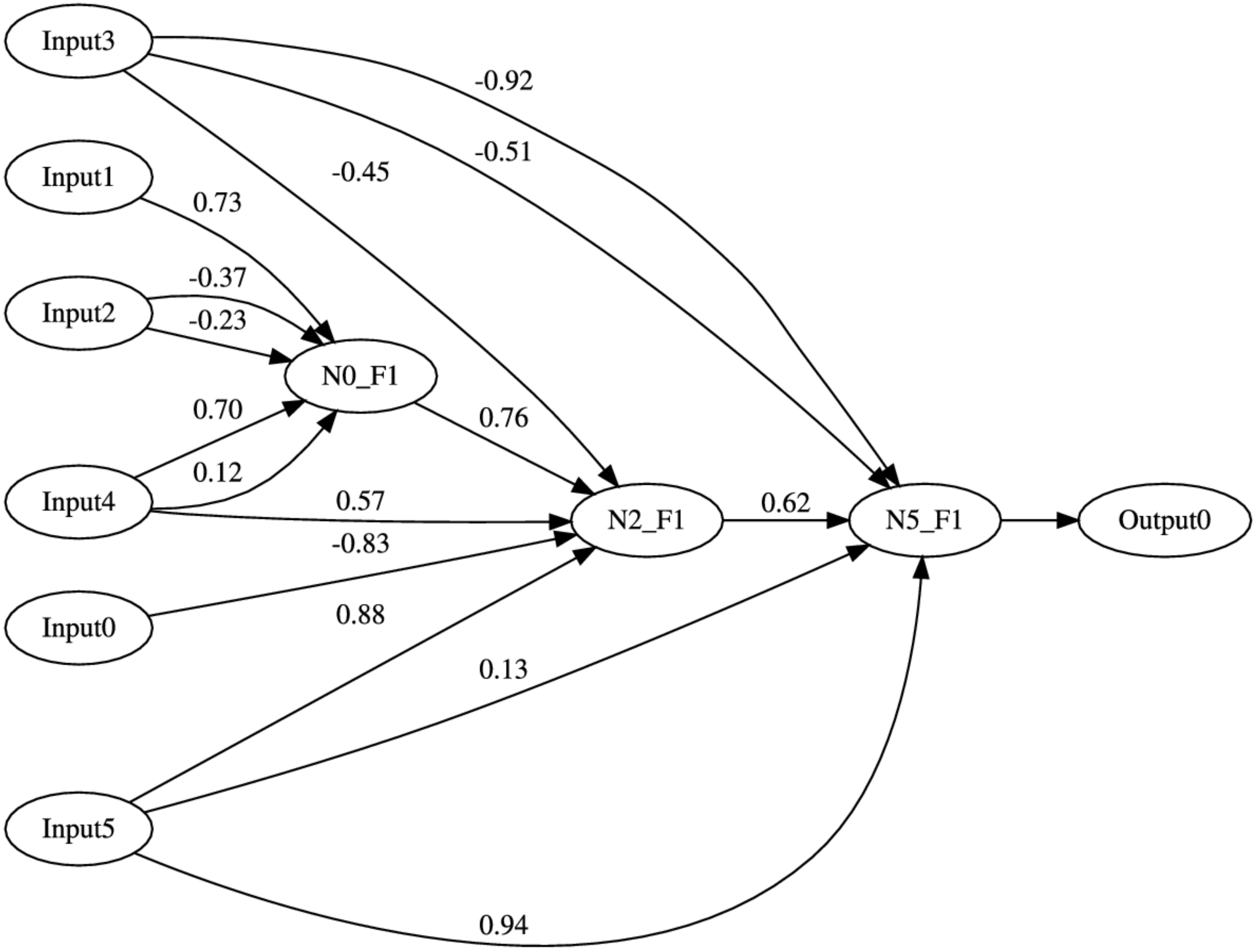


- Inactive Nodes
- Active Nodes



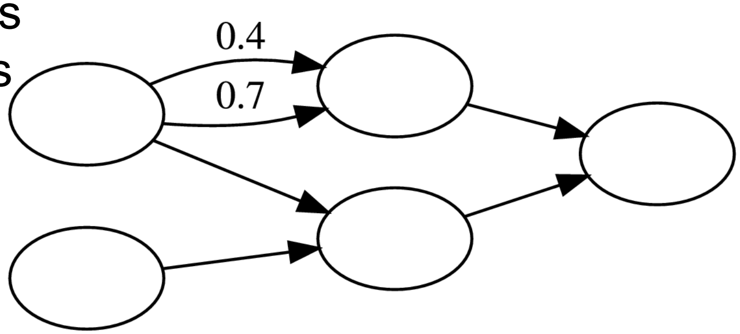
Andy Turner at568@york.ac.uk

Example ANN

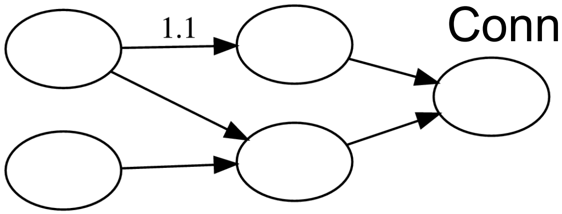

Input3
Input1
Input2
Input4
Input0
Input5
N0_F1
N2_F1
N5_F1
Output0
-0.92
-0.51
-0.45
0.73
-0.37
-0.23
0.70
0.12
0.76
0.57
-0.83
0.88
0.13
0.62
0.94

Multiple Connections

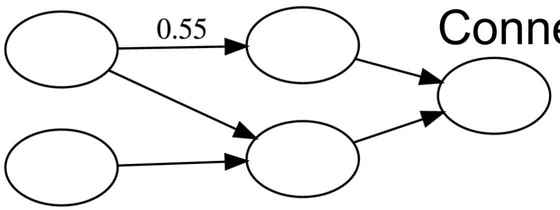
Multiple Connections between two nodes



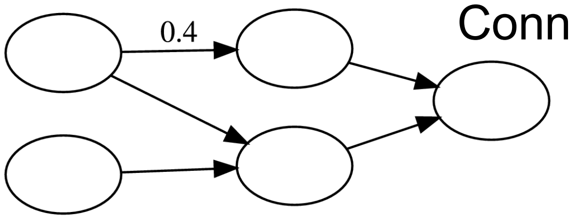
Allow Multiple Connections



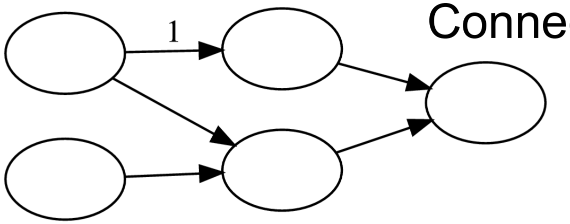
Average Connections



Only use first Connection



Cap Connection



Benchmarks

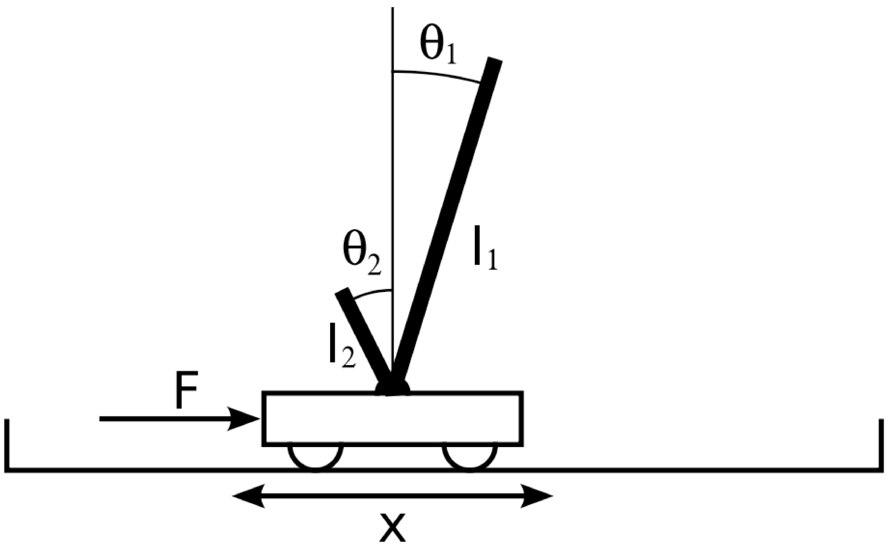
Benchmarks
- Double Pole Balancing
- Ball Throwing
- Proben1: Cancer1

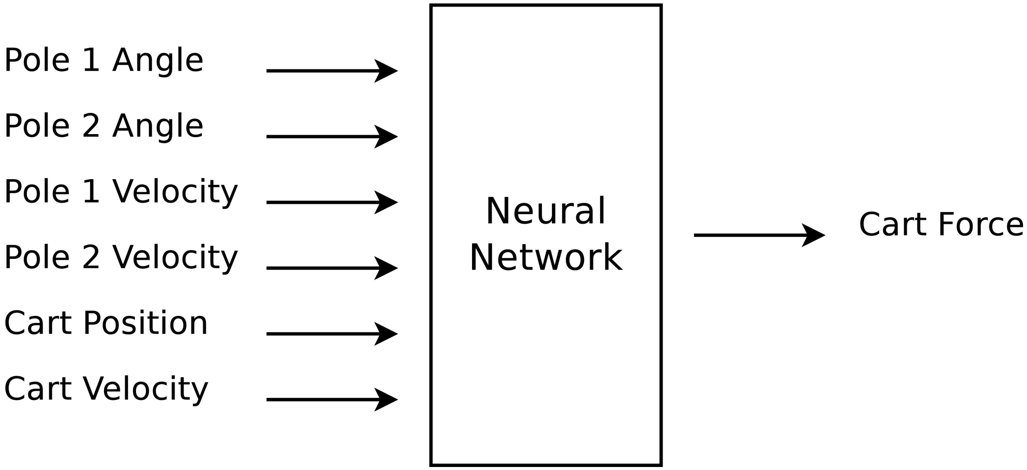
Parameters
- (1+4)-ES
- Uniform Mutation
- No Crossover
- Only Bipolar or Unipolar Sigmoid
- Allowed multiple connections between nodes

Comparison
- Can only use averages
- Data not available for statical significance tests

THE UNIVERSITY *of York*

Double Pole Balancing

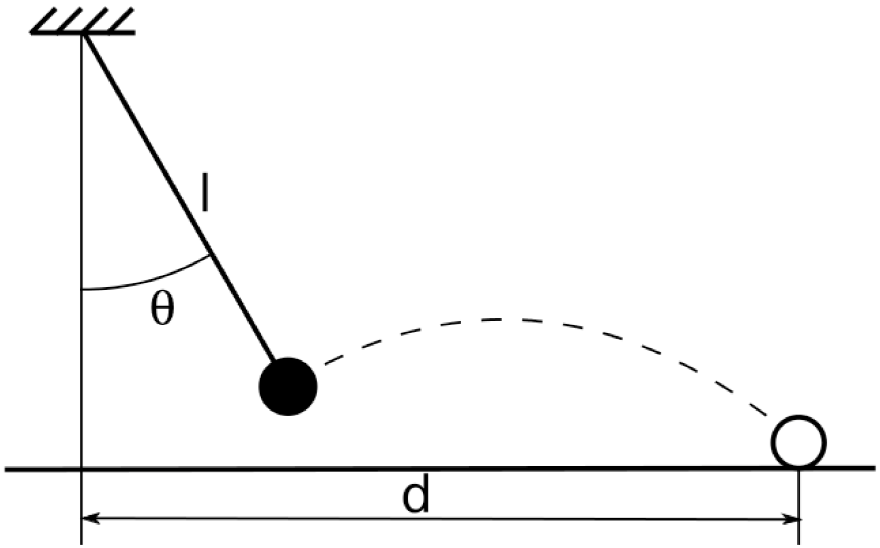




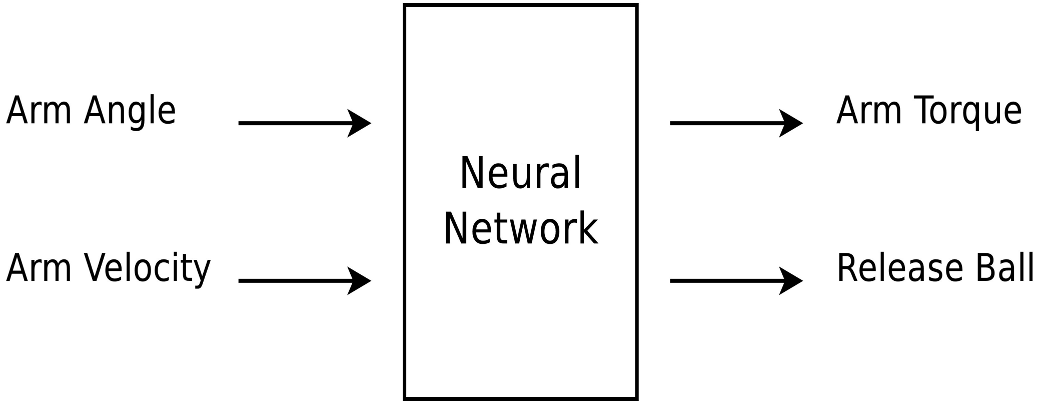
Method	Evaluations
DirE	410
CMA-ES	859
CoSyNE	954
CGPANN	**1111**
NEvA	2177
NEAT	3578
ESP	3800
Q-MPL	10583
SAIN	12600
EuSAIN	~19000
CNE	22100
CE	34000
EP	307200

Andy Turner
at568@york.ac.uk

Ball Throwing

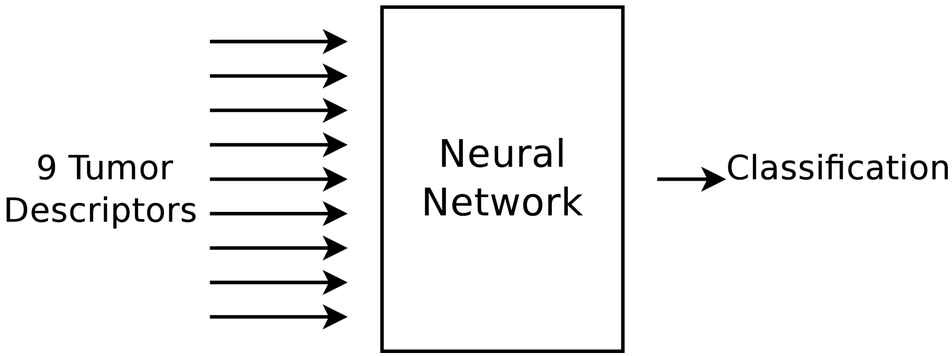


Method	Evaluations
CGPANN	**6069**
Compressed CoSyNE	8220
CoSyNE	10224



Andy Turner
at568@york.ac.uk

Proben1: Cancer1



Method	Train Err %	Test Err %
MFN	-	1.38
M-RAN	-	1.72
CGPANN	**2.68**	**1.89**
GA-MOO-ANN	-	1.9
MFNNCA	24.86	2
ACS	-	2.184
BP	-	3.506
CMAC ANN	0.59	3.94

Data from University of Wisconsin Hospital
(O. Mangasarian et al, 1990)

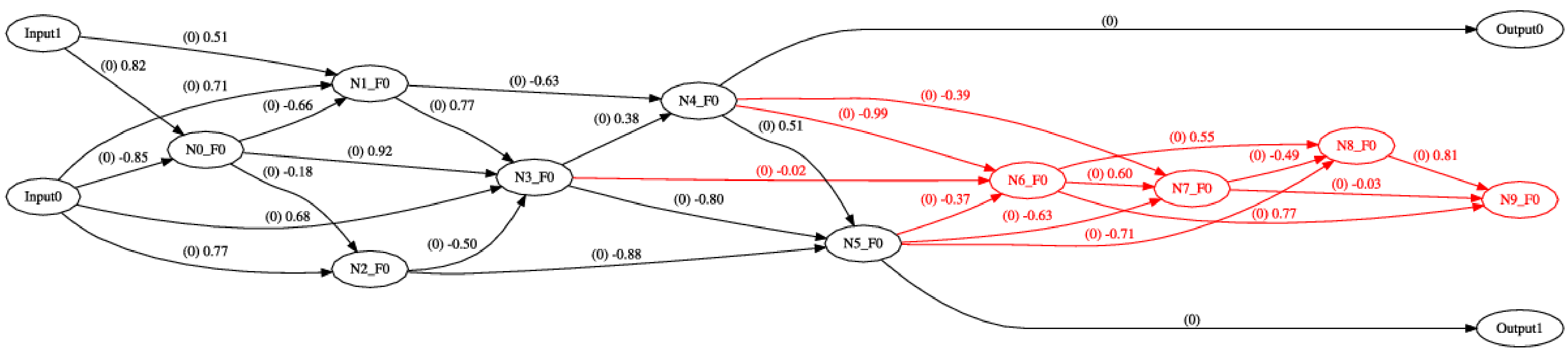
Following the Proben1 Document
(L. Prechelt, 1994)

Conclusion

Overall:
CGPANN is a highly competitive NeuroEvolutionary strategy which assumes very little about the structure of the neural network to be evolved.

Take Home Message:
CGPANN directly evolves the weights, number of nodes, topology, and nodes function of artificial neural networks. CGPANN also indirectly evolves the arity of each node. Additionally CGPANN has all of the benefits of CGP; natural resilience to bloat and neutrality in the genotype aiding evolution through genetic drift.

Questions



Andy Turner

at568@york.ac.uk