UNIVERSITY of York

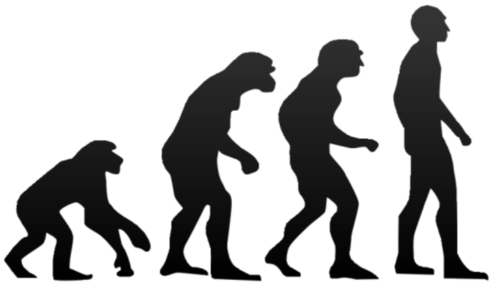
NeuroEvolution: The Importance of Transfer Function Evolution and Heterogeneous Networks

Andrew James Turner & Julian Francis Miller

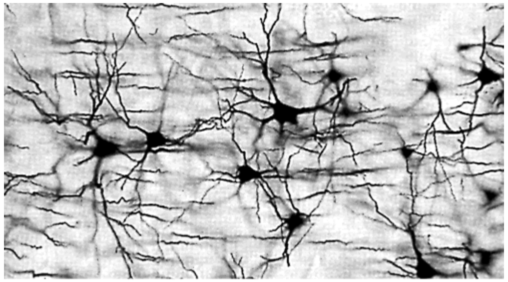
andrew.turner@york.ac.uk, julian.miller@york.ac.uk

NeuroEvolution: Application of Evolutionary Algorithms to Artificial Neural Networks

- Advantages
  - No restraint on topology
  - Can escape local optima
  - Applicable to reinforcement learning
  - **No restraint on transfer functions**



+



Main Topic

- The majority of NeuroEvolutionary methods create **homogeneous** networks.
- However NeuroEvolution can easily create **heterogeneous** networks.
- But do **heterogeneous** networks provide any benefit for NeuroEvolution?

There are two methods for evolving heterogeneous networks:

1) Allow evolution to select each neuron's transfer function from a predetermined set
2) Allow evolution to optimise parameters associated with each neuron's transfer function

(or a mixture of the two)

Key Questions

1) Does the choice of transfer function impact the training of homogeneous networks?
2) Does allowing evolution to select each neuron's transfer function produce better results than the homogeneous networks?
3) Does allowing evolution to optimise parameters associated with each neuron's transfer function produce better results than their non-parameterised counterparts?

NeuroEvolutionary Methods

Conventional NeuroEvolution

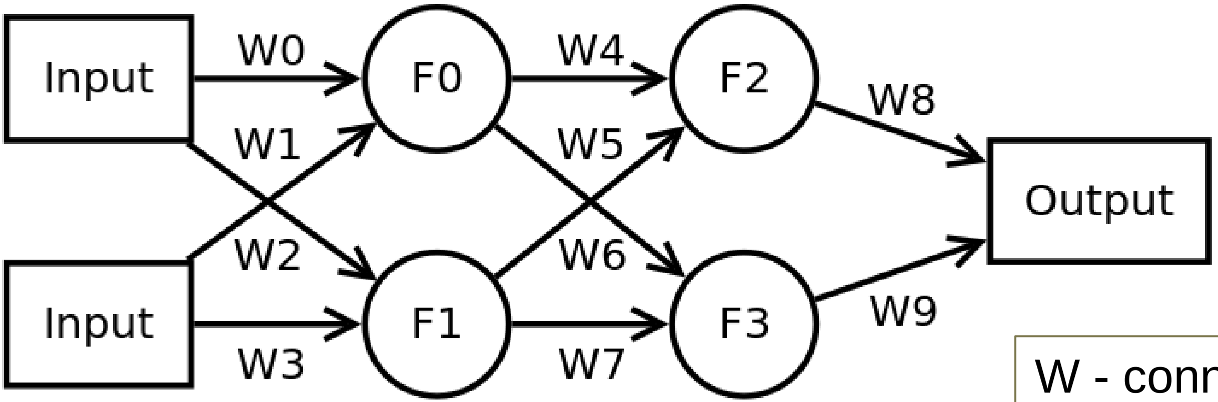
- Simplest and oldest (1990’s)
- Based on a Genetic Algorithm
- Evolves connection weights
- Fixed user defined topology

Cartesian Genetic Programming Artificial Neural Networks

- More complex and modern (2013)
- Based on Cartesian Genetic Programming
- Evolves connections weights
- Evolves topology (feed-forward and recurrent)

Conventional NeuroEvolution

- For a given topology:



W - connection weight
F - transfer function

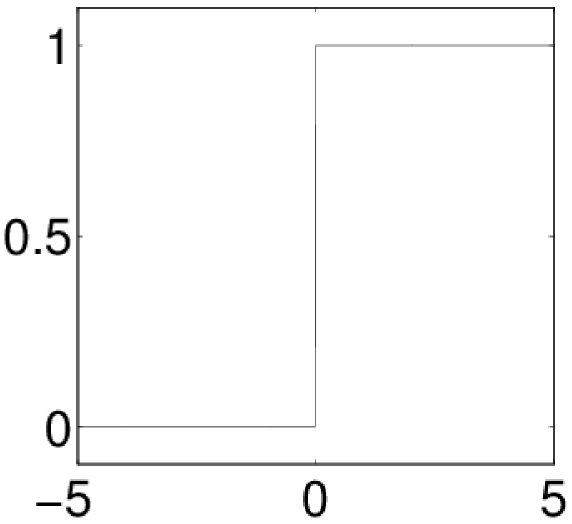
- Each chromosome takes the form:

{W1, W2, W3, W4, W5, W6, W7, W8, W9}

+ {F0, F1, F2, F3}

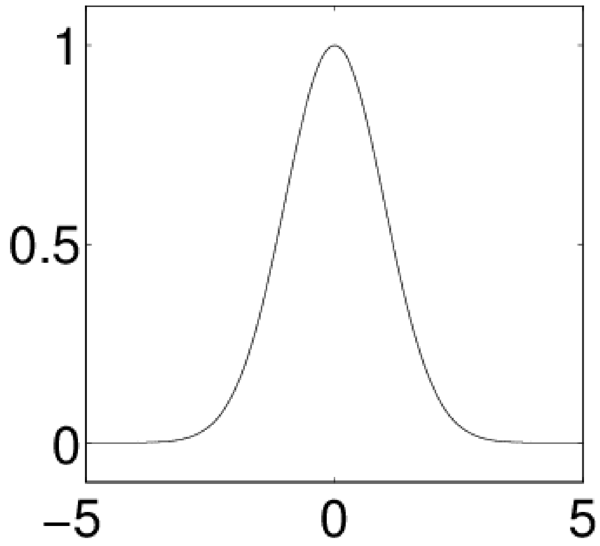
Transfer Functions

Heaviside Step



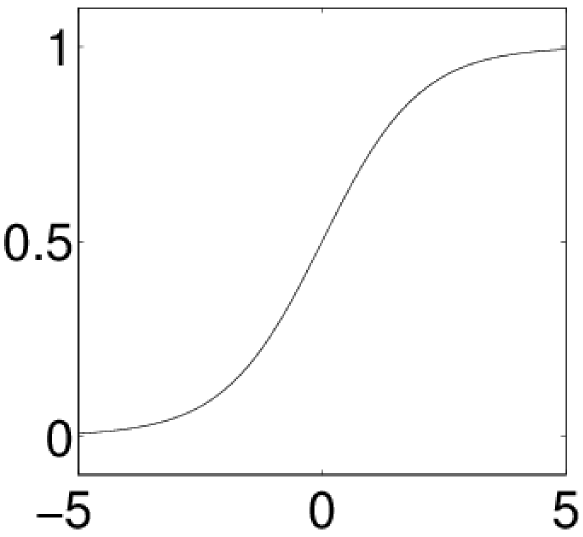
$$f(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Gaussian



$$f(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Logistic

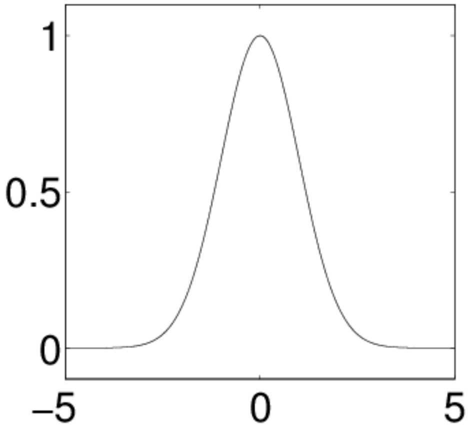


$$f(x) = \frac{1}{1 + \exp(-\sigma x)}$$

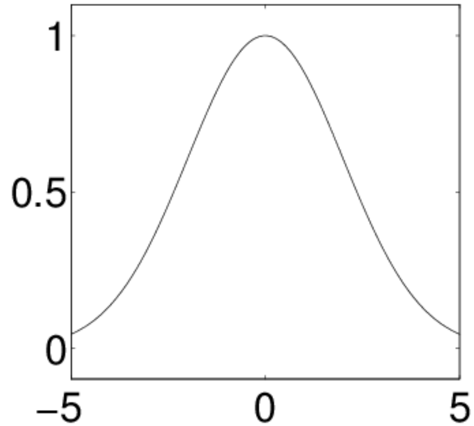
$\sigma = 1$

Parameterised Transfer Functions

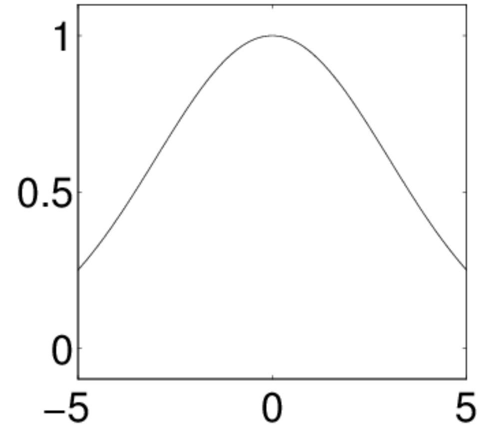
Gaussian



σ = 1

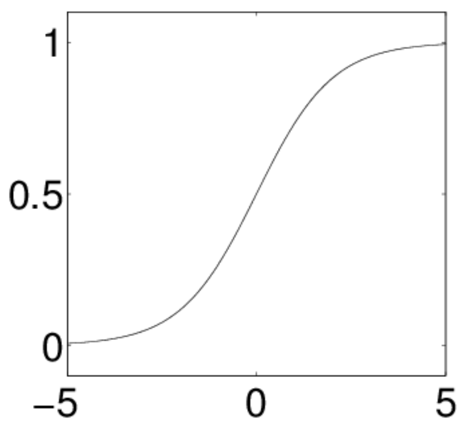


σ = 2

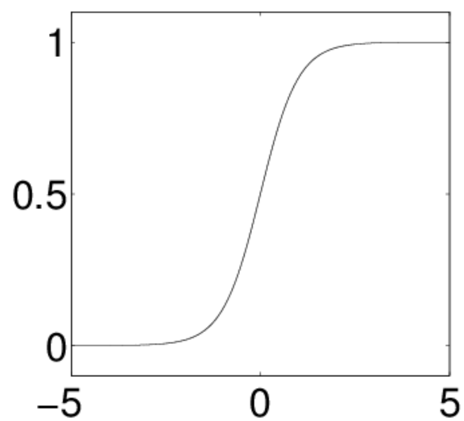


σ = 3

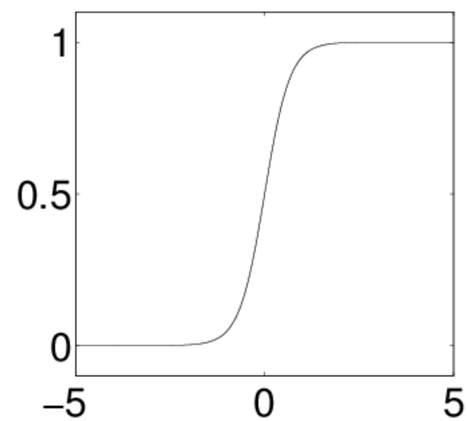
Logistic



σ = 1



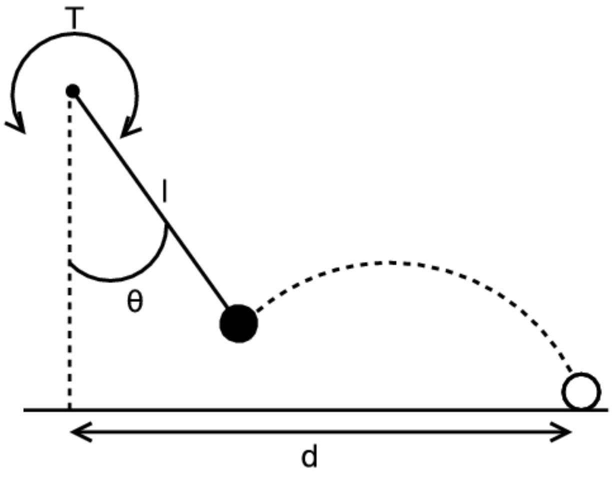
σ = 2



σ = 3

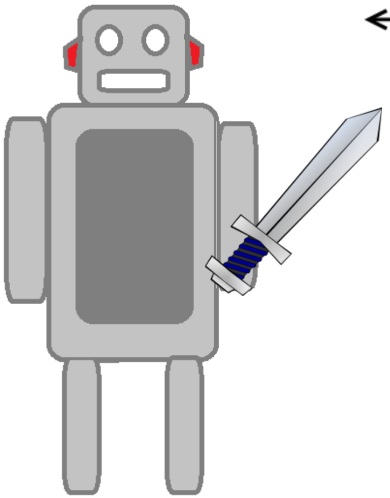
Reinforcement Learning

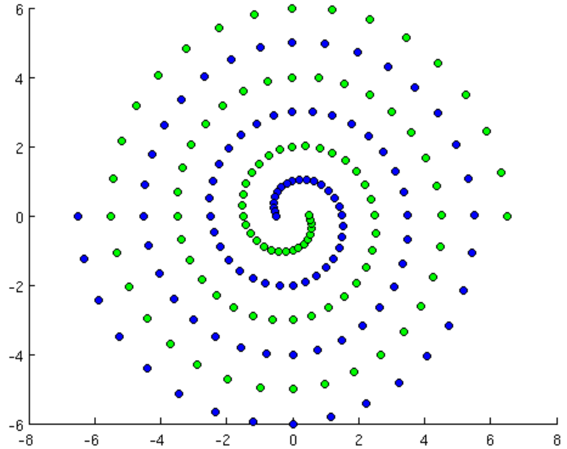
- Ball Throwing



Supervised Learning

- Full Adder
- Monks Problem 1
- Two Spirals
- Proben1: Cancer1





NeuroEvolution Parameters

General Parameters

- (1+4)-ES
- 3% probabilistic mutation
- No Crossover
- Connection weight range +/- 5
- 1000 generations (4001 evaluations)
- Average fitness from 50 runs

Conventional NeuroEvolution

- 3 fully connected hidden layers containing 10 nodes

Cartesian Genetic Programming Artificial Neural Networks

- Maximum of 30 nodes each with a maximum arity of 10

Experiment 1

- Does the choice of transfer function impact the training of homogeneous networks?

Benchmark	Target	Step	Gaussian	Logistic
Ball Throwing	(↑) 9.50	5.63	**6.41**	5.57
Full Adder	(↑) 16.00	**16.00**	15.92	15.86
Monks (train)	(↓) 0.00	**9.82**	27.65	11.03
Monks (test)	(↓) 0.00	27.98	43.16	**25.87**
Two Spirals	(↓) 0	70.00	**56.54**	81.52
Cancer1 (train)	(↓) 0.00	10.50	5.44	**3.35**
Cancer1 (test)	(↓) 0.00	14.44	7.49	**3.54**

Conventional NeuroEvolution

Experiment 1

- Does the choice of transfer function impact the training of homogeneous networks?

Benchmark	Target	Step	Gaussian	Logistic
Ball Throwing	(↑) 9.50	**9.34**	7.34	5.80
Full Adder	(↑) 16.00	**15.94**	15.40	15.78
Monks (train)	(↓) 0.00	**10.71**	15.27	12.72
Monks (test)	(↓) 0.00	**13.44**	21.93	18.79
Two Spirals	(↓) 0	67.42	**66.36**	80.64
Cancer1 (train)	(↓) 0.00	**2.16**	2.55	2.50
Cancer1 (test)	(↓) 0.00	2.71	2.74	**2.09**

Cartesian Genetic Programming Artificial Neural Networks

Experiment 1

Results

- The choice of neuron transfer function clearly effects homogeneous networks
- Different transfer functions are suited to different tasks
- It is not known in advance which transfer function will most suitable
- The Heaviside step function did surprisingly well.

Experiment 2

- Does allowing evolution to select each neuron's transfer function produce better results than the homogeneous networks?

Benchmark	Target	Average Homogeneous	Heterogeneous Networks
Ball Throwing	(↑) 9.50	5.87	**8.83**
Full Adder	(↑) 16.00	15.93	**16.00**
Monks (train)	(↓) 0.00	**16.17**	16.87
Monks (test)	(↓) 0.00	**32.34**	33.69
Two Spirals	(↓) 0	96.35	**63.46**
Cancer1 (train)	(↓) 0.00	6.43	**3.87**
Cancer1 (test)	(↓) 0.00	8.49	**5.16**

Conventional NeuroEvolution

Experiment 2

- Does allowing evolution to select each neuron's transfer function produce better results than the homogeneous networks?

Benchmark	Target	Average Homogeneous	Heterogeneous Networks
Ball Throwing	(↑) 9.50	7.49	**8.90**
Full Adder	(↑) 16.00	**15.71**	15.68
Monks (train)	(↓) 0.00	12.90	**11.02**
Monks (test)	(↓) 0.00	18.05	**16.72**
Two Spirals	(↓) 0	71.47	**70.24**
Cancer1 (train)	(↓) 0.00	2.40	**2.33**
Cancer1 (test)	(↓) 0.00	**2.51**	2.69

Cartesian Genetic Programming Artificial Neural Networks

Experiment 2

Results

- Heterogeneous networks outperformed the average homogeneous network
- Therefore, if the optimal transfer function is not known, evolving heterogeneous networks produces better results on average than a random choice of transfer function

UNIVERSITY of York

Experiment 3

- Does allowing evolution to optimise parameters associated with each neuron's transfer function produce better results than their non-parameterised counterparts?

Benchmark	Target	Regular Gaussian	Parameterised Gaussian
Ball Throwing	(↑) 9.50	6.41	**8.15**
Full Adder	(↑) 16.00	15.92	**15.96**
Monks (train)	(↓) 0.00	27.65	**26.24**
Monks (test)	(↓) 0.00	43.16	**41.99**
Two Spirals	(↓) 0	**56.54**	66.26
Cancer1 (train)	(↓) 0.00	5.44	**3.09**
Cancer1 (test)	(↓) 0.00	7.49	**3.53**

Conventional NeuroEvolution

University of York

Experiment 3

- Does allowing evolution to optimise parameters associated with each neuron's transfer function produce better results than their non-parameterised counterparts?

Benchmark	Target	Regular Logistic	Parameterised Logistic
Ball Throwing	(↑) 9.50	5.57	**6.21**
Full Adder	(↑) 16.00	15.86	**16.00**
Monks (train)	(↓) 0.00	11.03	**10.45**
Monks (test)	(↓) 0.00	**25.87**	27.00
Two Spirals	(↓) 0	81.52	**74.28**
Cancer1 (train)	(↓) 0.00	**3.35**	3.89
Cancer1 (test)	(↓) 0.00	**3.54**	4.79

Conventional NeuroEvolution

Experiment 3

- Does allowing evolution to optimise parameters associated with each neuron's transfer function produce better results than their non-parameterised counterparts?

Benchmark	Target	Regular Gaussian	Parameterised Gaussian
Ball Throwing	(↑) 9.50	7.34	**7.62**
Full Adder	(↑) 16.00	15.40	**15.72**
Monks (train)	(↓) 0.00	15.27	**15.26**
Monks (test)	(↓) 0.00	21.93	**21.59**
Two Spirals	(↓) 0	**66.36**	69.50
Cancer1 (train)	(↓) 0.00	2.55	**2.48**
Cancer1 (test)	(↓) 0.00	2.74	**2.31**

Cartesian Genetic Programming Artificial Neural Networks

Experiment 3

- Does allowing evolution to optimise parameters associated with each neuron's transfer function produce better results than their non-parameterised counterparts?

Benchmark	Target	Regular Logistic	Parameterised Logistic
Ball Throwing	(↑) 9.50	5.80	**7.82**
Full Adder	(↑) 16.00	**15.78**	15.74
Monks (train)	(↓) 0.00	12.72	**10.07**
Monks (test)	(↓) 0.00	18.79	**17.26**
Two Spirals	(↓) 0	80.64	**75.60**
Cancer1 (train)	(↓) 0.00	2.50	**2.42**
Cancer1 (test)	(↓) 0.00	**2.09**	2.28

Cartesian Genetic Programming Artificial Neural Networks

Experiment 3

Results

- Optimizing parameters associated with each neurons transfer function produces better results, on average, than their non-parameterised counterparts.

Conclusion

- NeuroEvolution can be used to create heterogeneous neural networks
- There are two, mutually inclusive, methods for allowing NeuroEvolution to create heterogeneous networks
- Both of these methods have been shown, on average, to outperform homogeneous networks
- Both of these methods are likely compatible with all NeuroEvolutionary techniques

*All results were also analysed using the Mann-Whitney U-test and effect size statistics

Mini Plug

CGP-Library

- Cartesian Genetic Programming
- NeuroEvolution
- Simple & extendible
- Written in C
- Open source (LGPL)

http://andrewjamesturner.co.uk/
https://github.com/AndrewJamesTurner/CGP-Library

Questions?