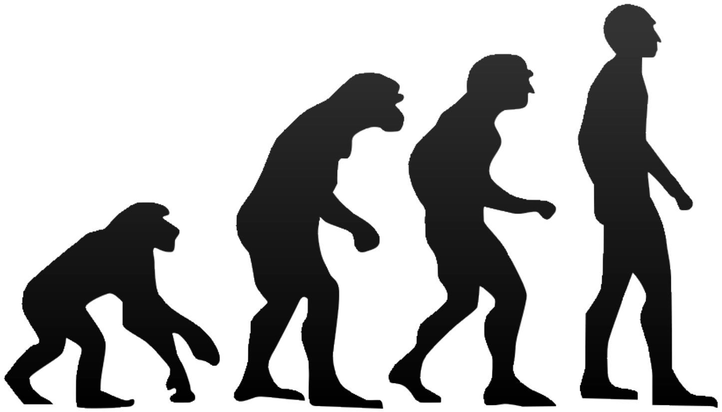
The University of York

Bio-Inspired Computing

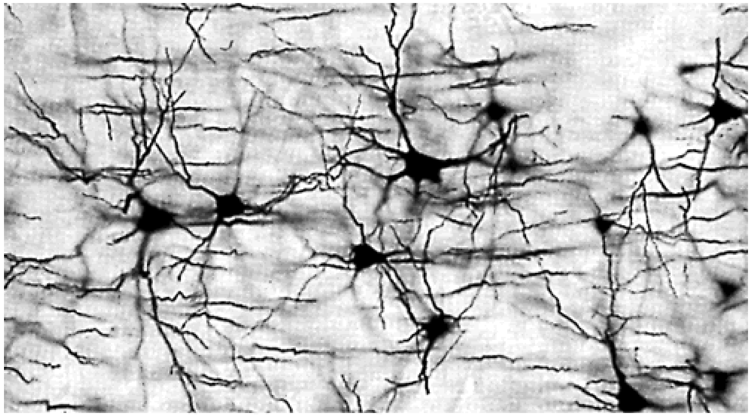
Lecture 10: CGP & Neural Networks

Andy Turner

NeuroEvolution



+



Evolutionary Computation

Neural Networks

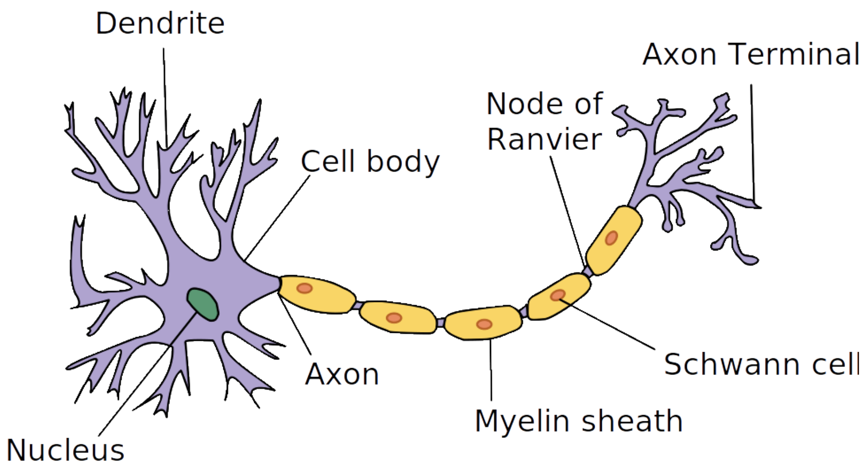
Biological Neural Networks

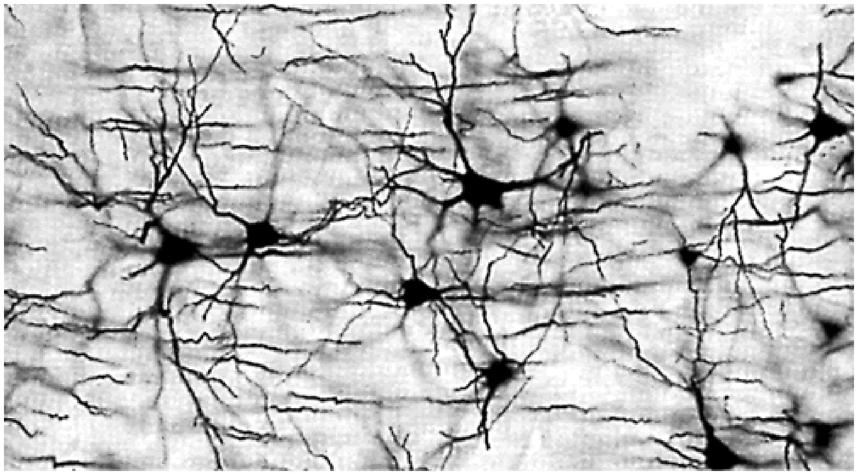
- Structure of biological brain
- Responsible for our own intelligence
- Very different to electronic computation
- Highly fault tolerant
- Parallel computation

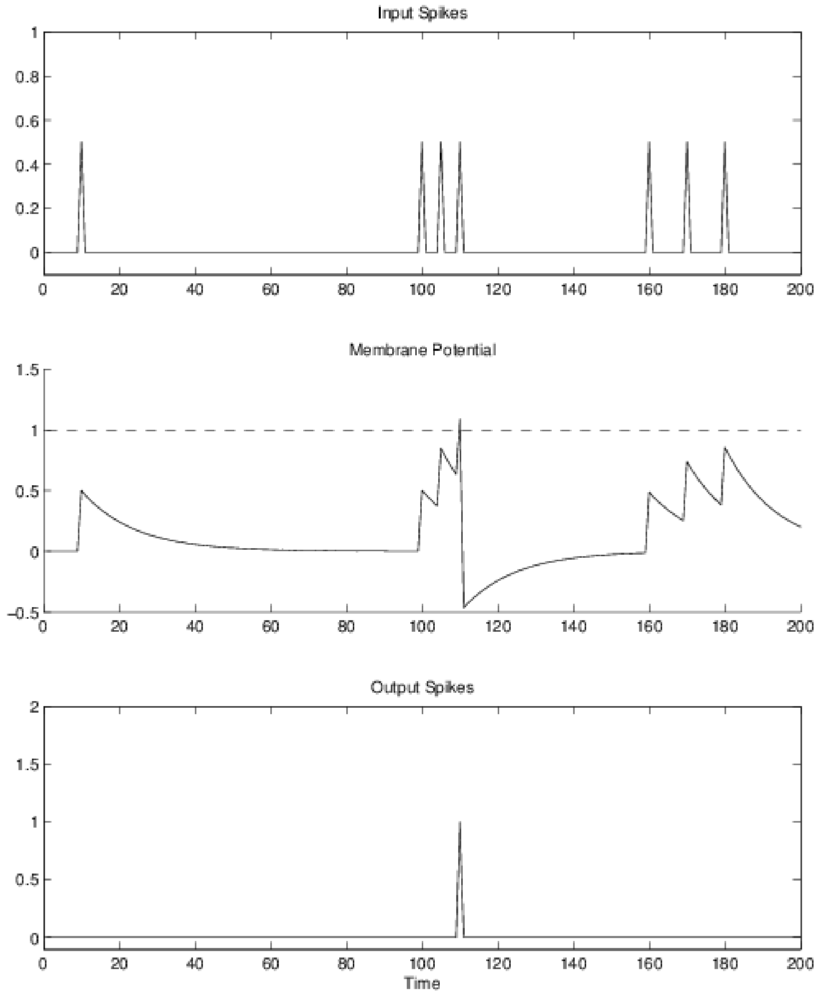
Brain Vs Computers

	Human Brain	Intel's Quad-Core + GPU Core i7
Functional element	Neuron	Transistor
Num elements	~ 8.6 x10^10	1.6 x10^9
Num Inputs	~ 7000 (avg)	2
Num Outputs	1	1
Num Connections	~ 6 x10^14	3.2 x10^9
Power Consumption	~ 20 W	200 W (under load)
Centralised Control	No	Yes
High fault tolerance	Yes	No
Data Storage	Distributed memory	Dedicated memory
Processing	Highly parallel	Sequential

Biological Neuron







History of Artificial Neural Networks

1896 - Discovery of biological neuron

1943 - First artificial neuron model

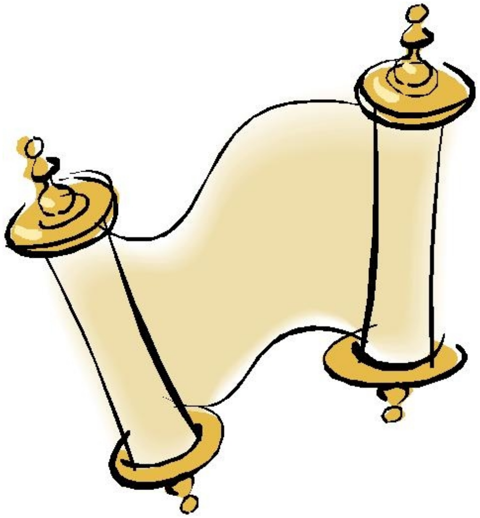
1958 - Mark I Perceptron

Dark Ages...

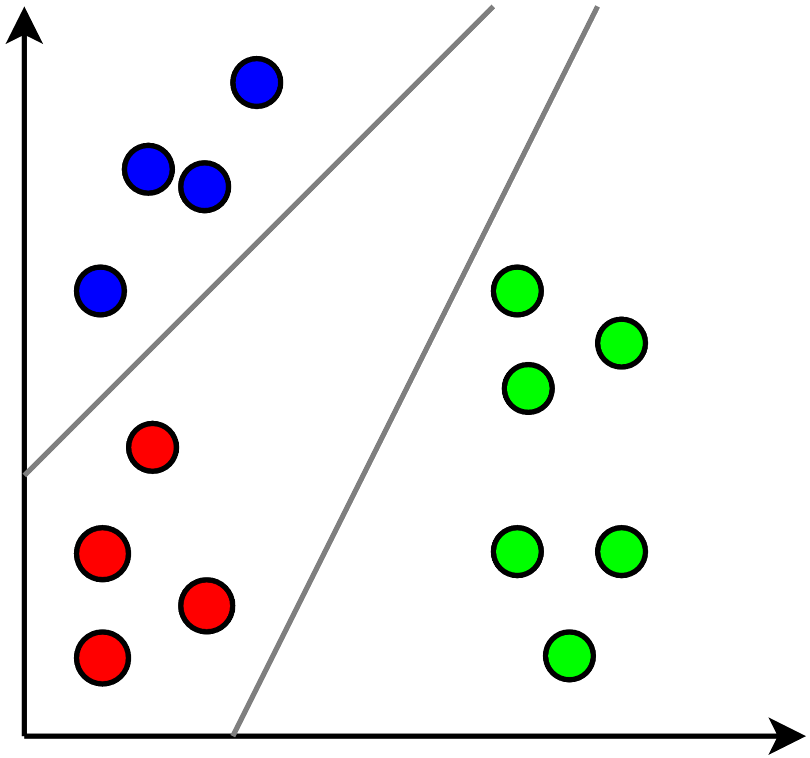
1986 - Back Propagation

1990 - NeuroEvolution

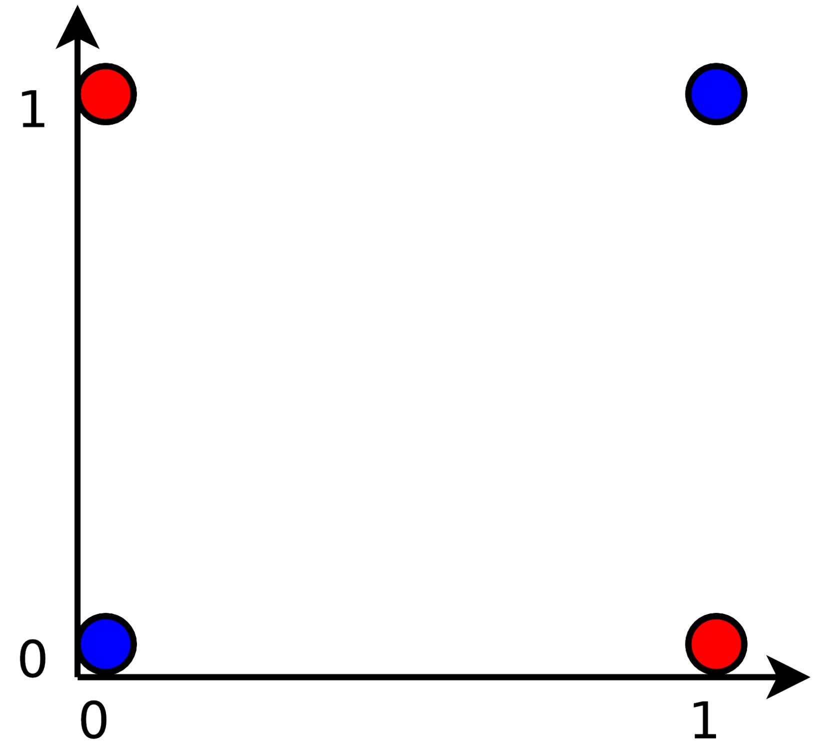
2013 - Still active area of research



Linearly Separable

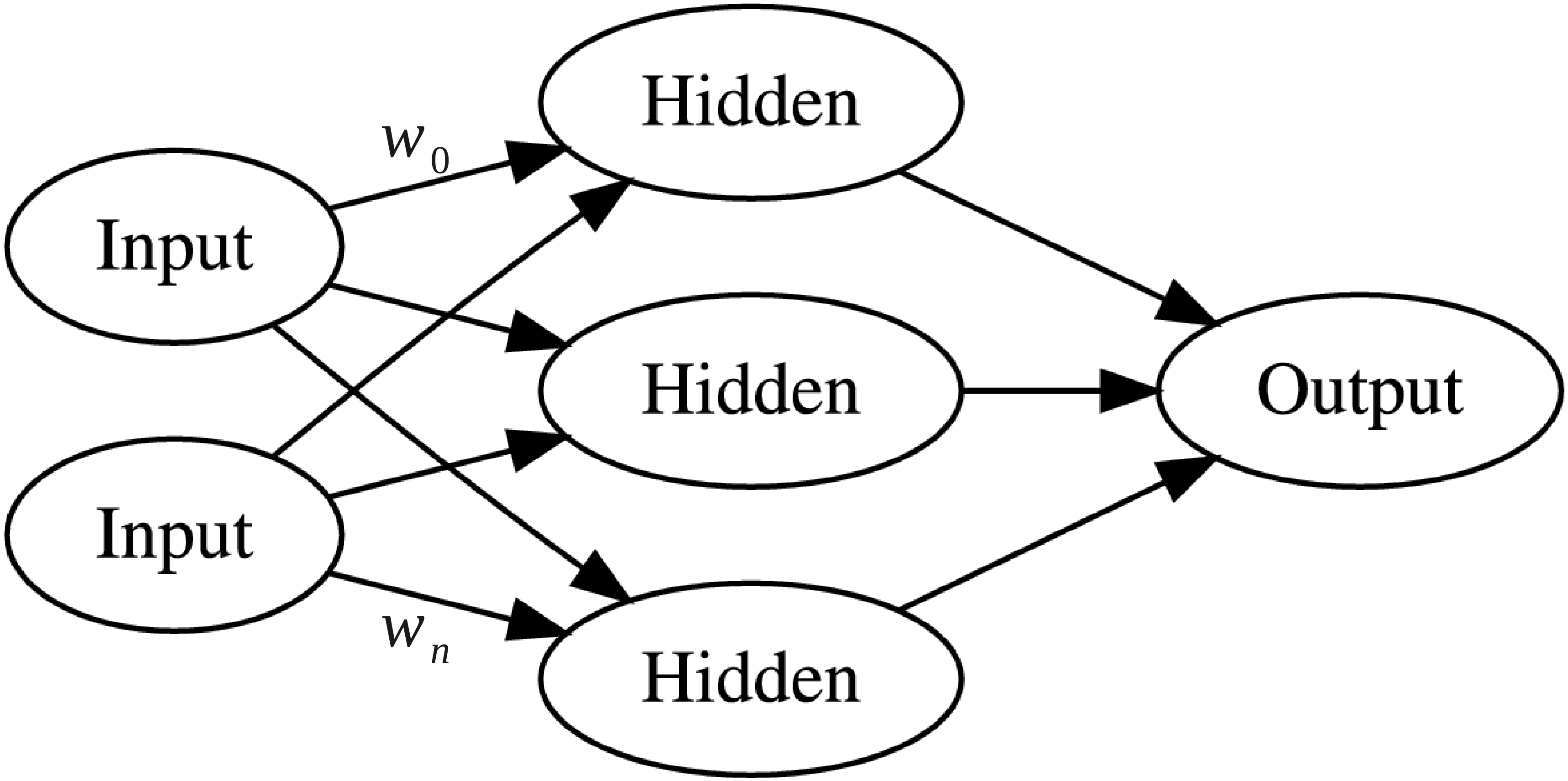


Linearly Separable



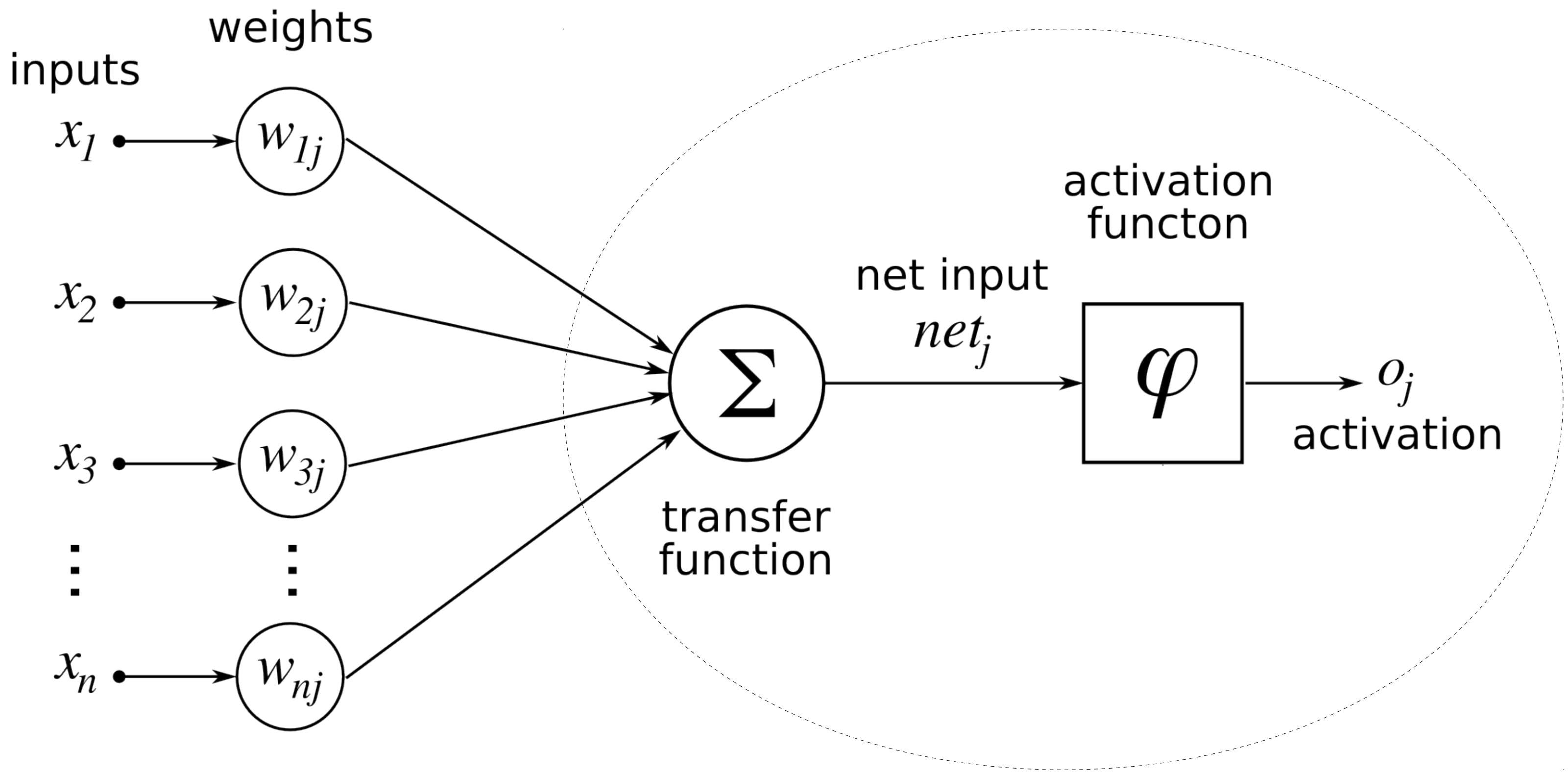
Not Linearly Separable
XOR Gate

Artificial Neural Network



Input Layer | Hidden Layer(s) | Output Layer

Artificial Neurons

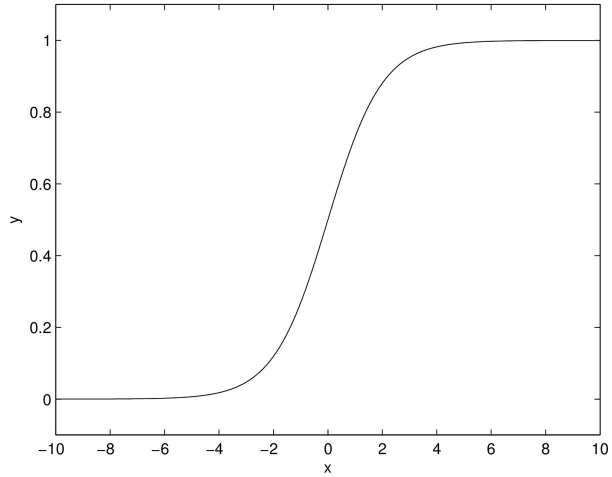


$$net = \sum_{i=1}^{n} w_i x_i \qquad o = \rho(net)$$

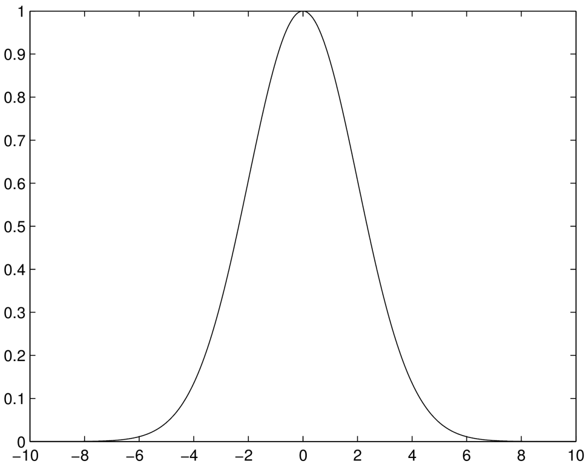
- Activation Functions

$$net = \sum_{i=1}^{n} w_i x_i$$

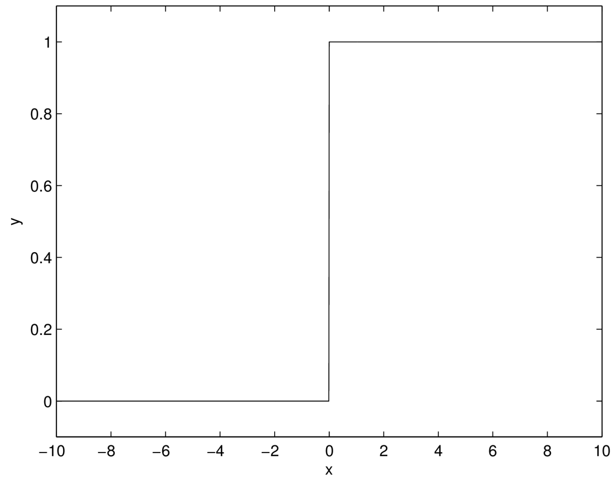
$$o = \rho(net)$$



Logistic Function (sigmoid)



Gaussian Function

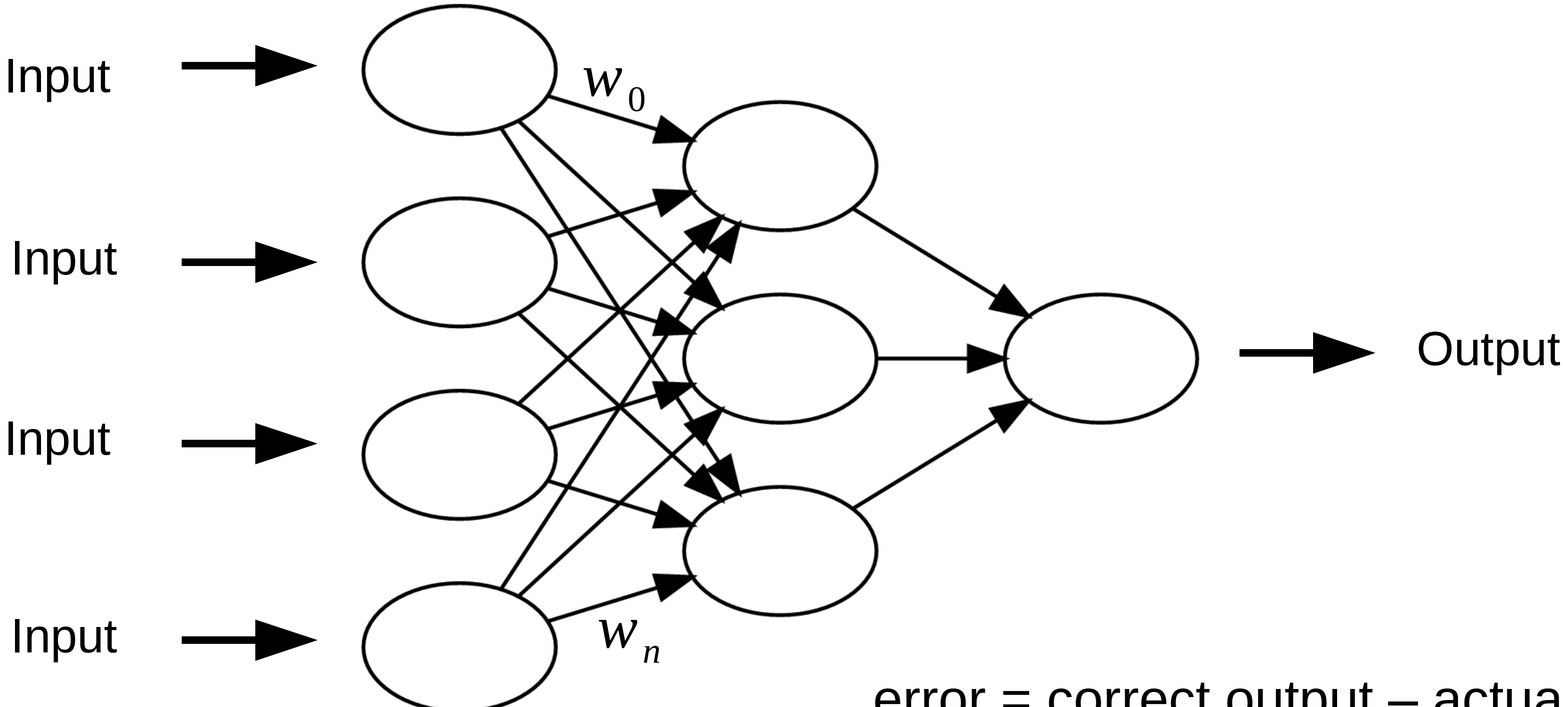


Step Function

Neural Network Training Methods

- **Back Propagation – Most Popular**
- Hopfield Networks
- Bolzmann Machine
- Radial Basis Function Networks
- Deep Belief Networks
- Hebbian Learning
- **NeuroEvolution – My Interest**

Back Propagation



error = correct output – actual output

$$\frac{\delta w_i}{\delta error} = \&\#£*!\#$$

$$w_i = w_i + \frac{\delta w_i}{\delta error}$$

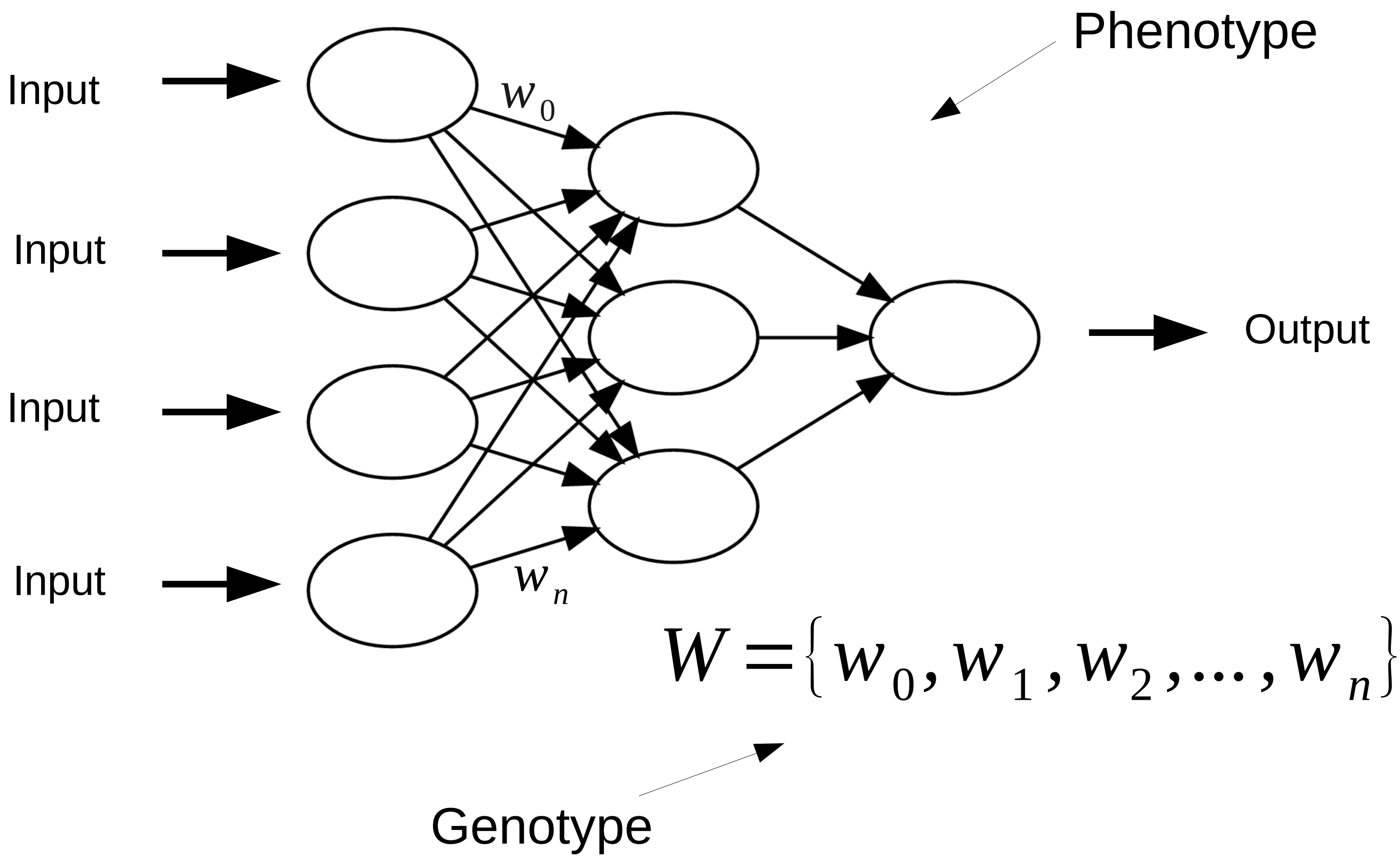
Back Propagation

- **Pros :)**
  - Saved neural networks from “dark ages”
  - Can solve linearly separable problems
- **Cons :(**
  - Neuron activation functions must differentiable
  - Neuron functions are typically homogeneous
  - Easily trapped in local optima (gradient decent)
  - Topology must be chosen in advance
  - Only suited to supervised learning
  - Not *directly* applicable to recurrent neural networks

NeuroEvolution

- **Conventional NeuroEvolution (CNE) – simplest & oldest**
- Symbiotic Adaptive NeuroEvolution (SAIN)
- Enforced SubPopulation (ESP)
- NeuroEvolution of Augmenting Topologies (NEAT)
- GeNeralized Acquisition of Recurrent Links (GNARL)
- Evolutionary Programming Artificial Neural Networks (EPNET)
- **CGP Artificial Neural Networks (CGPANN) – My Interest**

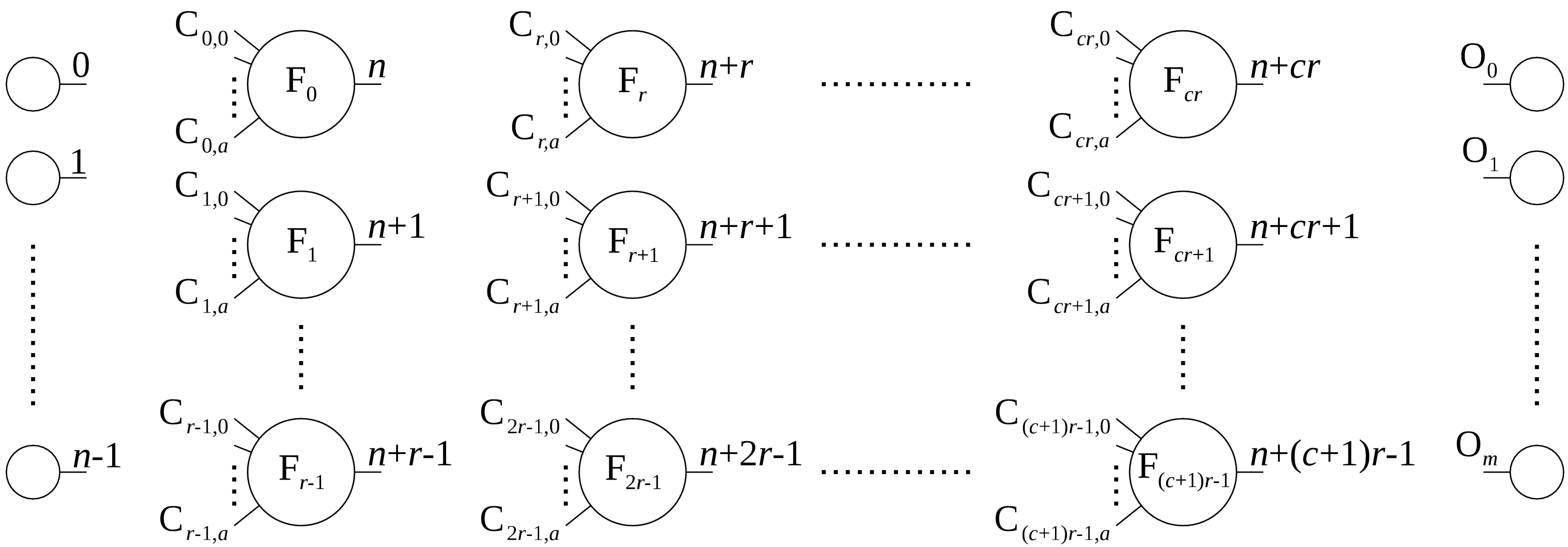
Conventional NeuroEvolution



Conventional NeuroEvolution

- **Pros :)**
  - No limitations on activation function
  - Less prone to local optima
  - Suited to supervised learning
  - Suited to reinforcement learning
  - Applicable to recurrent networks
- **Cons :(**
  - Topology still chosen in advance by user
  - Neuron functions chosen in advance by user

CGP Artificial Neural Networks

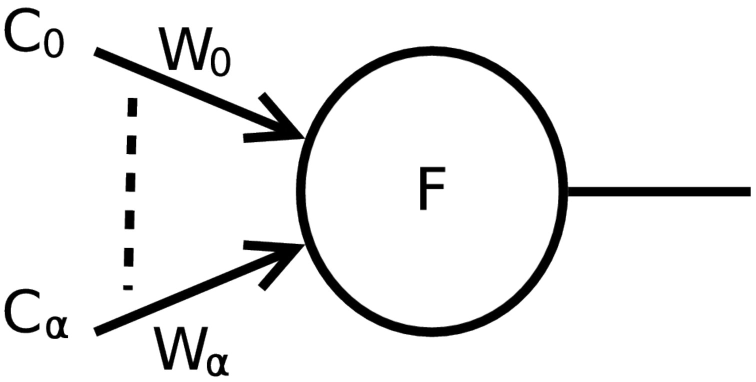


$$F_0 C_{0,0} \ldots C_{0,a}\, F_1 C_{1,0} \ldots C_{1,a} \ldots\ldots\ldots F_{(c+1)r-1}\, C_{(c+1)r-1,0} \ldots C_{(c+1)r-1,a}\, O_0 O_1 \ldots O_m$$

CGP Artificial Neural Networks

- Almost standard CGP
- Extra weight gene for each connection
- Use functions suited to Neural Networks
  - Heavy side step function
  - Sigmoid (logistic function)
  - Radial Basis Functions (Gaussian)
- High arity nodes (for high connectivity)

$$FC_0W_0C_1W_1 \ldots C_\alpha W_\alpha$$



CGP Artificial Neural Networks

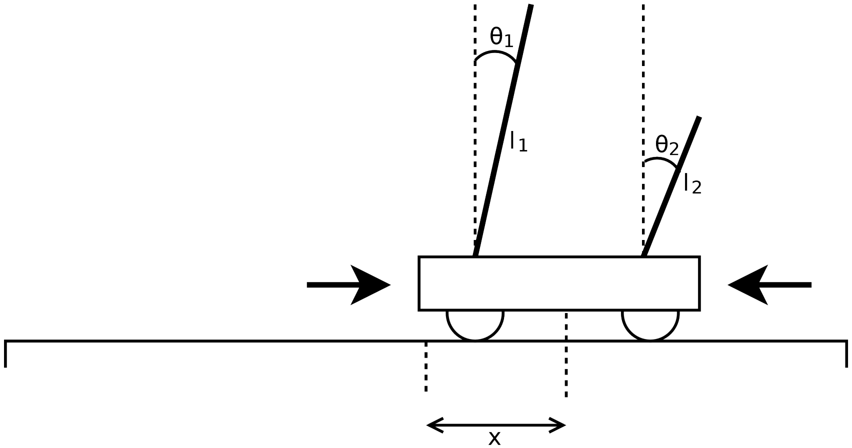
- All the benefits of conventional NeuroEvolution, and...
  - Also evolves topology (including recurrent)
  - Creates heterogeneous networks
- But does this actually help...

Evolving Topology

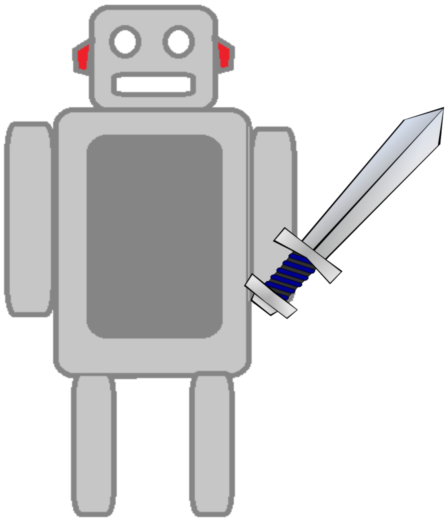
- It is known that the choice of topology drastically influences the effectiveness of back propagation.
- But is this true for Coventional NeuroEvolution?
- And can CGP NeuroEvolution be used to find suitable topologies?

Evolving Topology

- **Experiment!**
  - Conventional NeuroEvolution Vs CGPANN
  - Does the choice of topology matter?
  - Does evolving topology help?



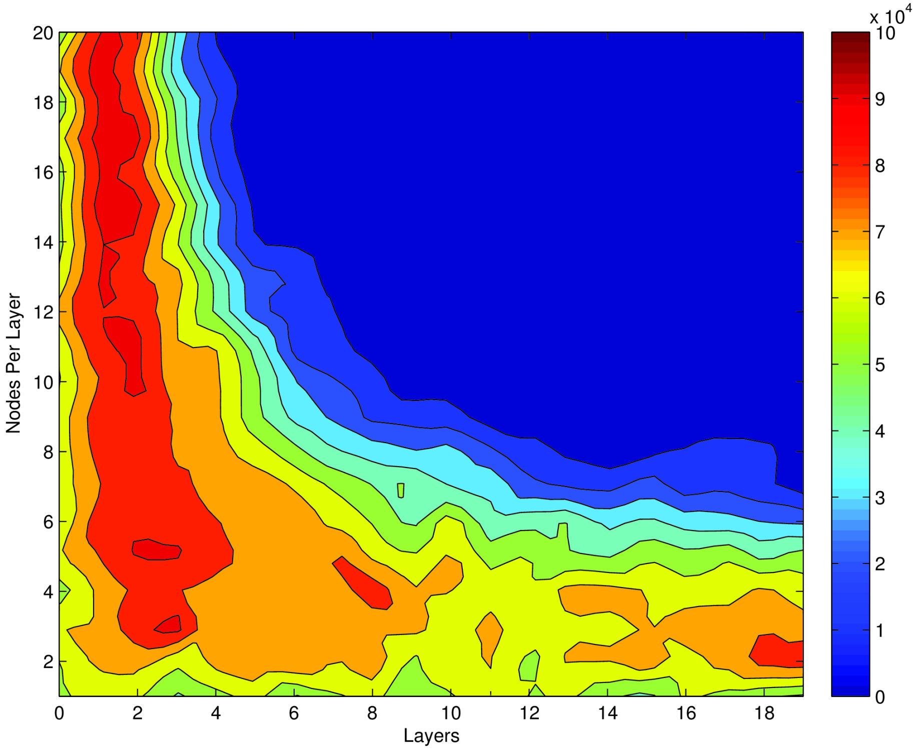
Double Pole Balancing

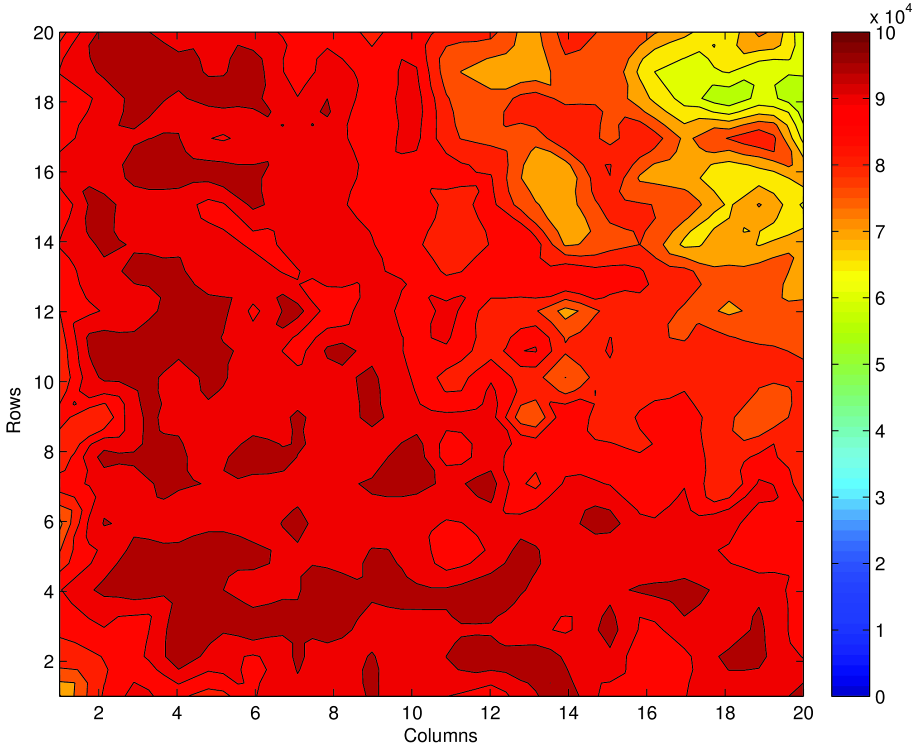


Monks Problem

THE UNIVERSITY of York

Double Pole Balancing



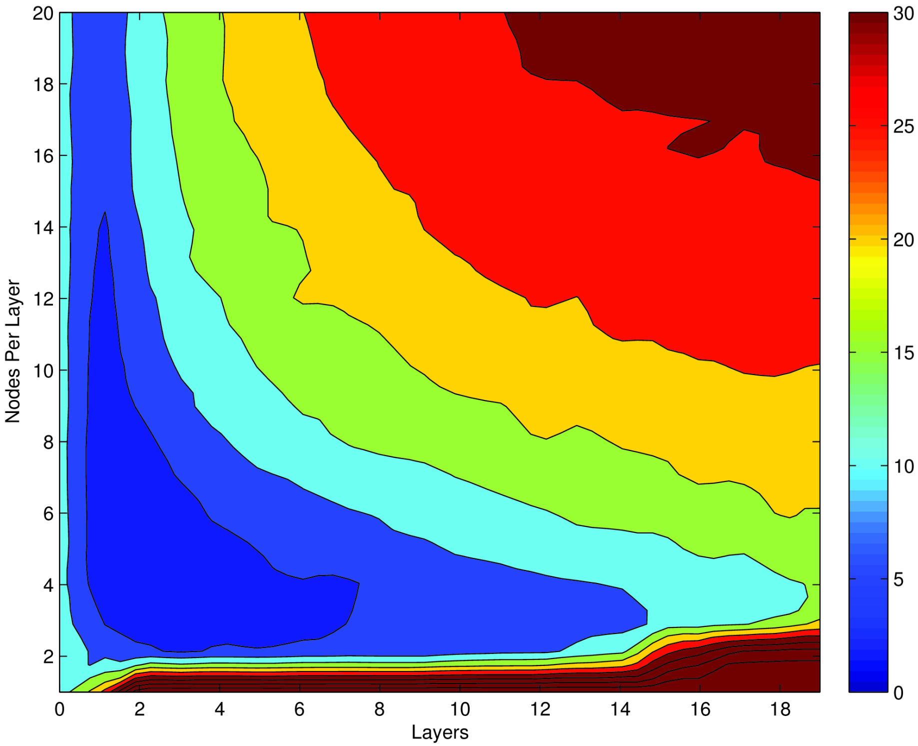


Conventional NeuroEvolution (Fixed Topology)

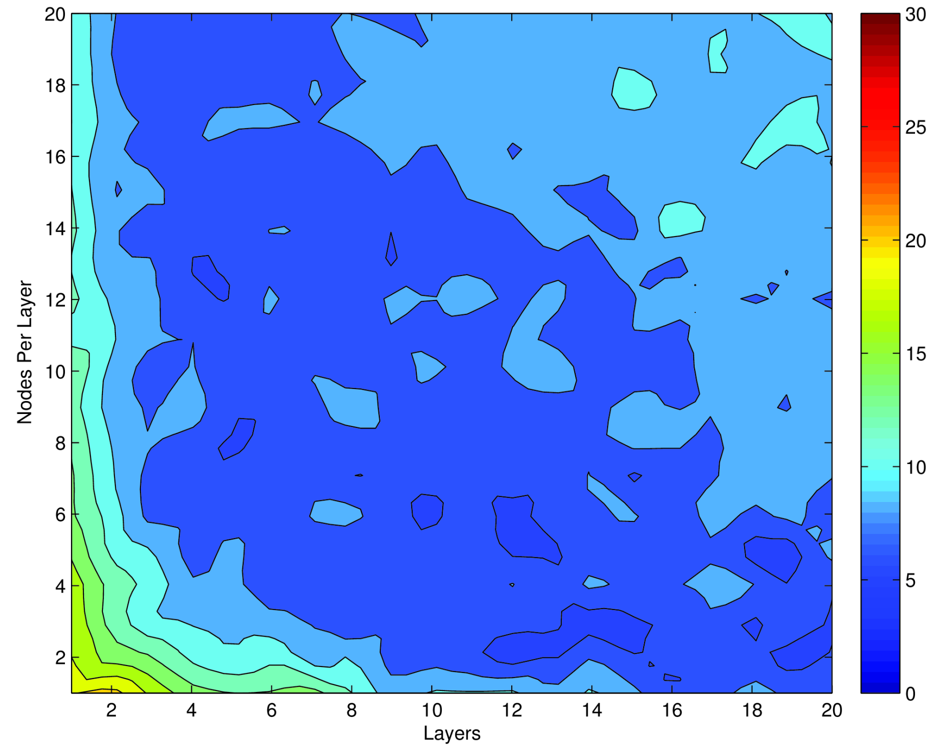
CGP Artificial Neural Networks (Evolved Topology)

Fitness is number of seconds poles were balanced for i.e. higher is better

Monks Problem



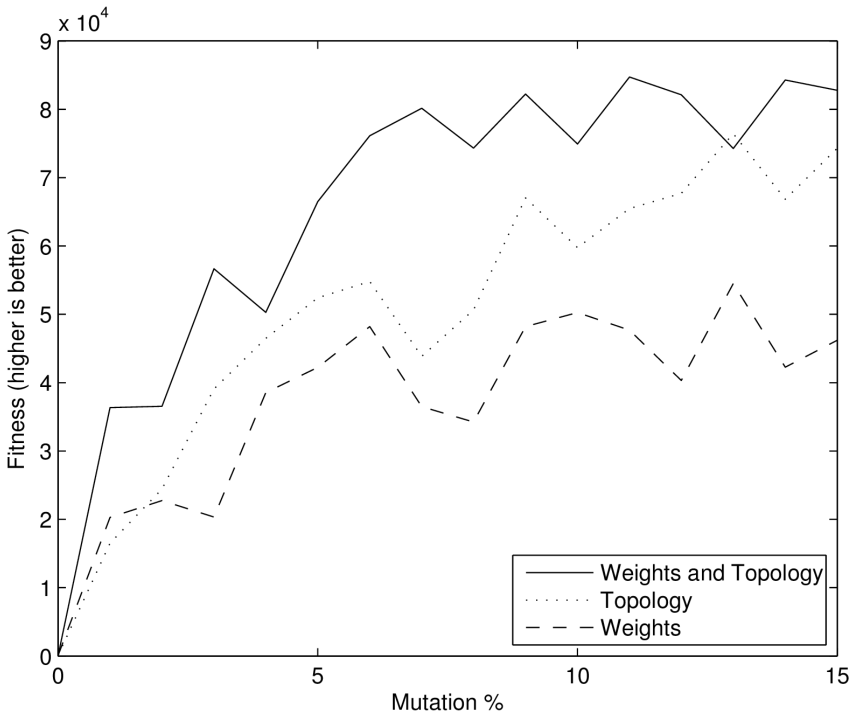
Conventional NeuroEvolution (Fixed Topology)



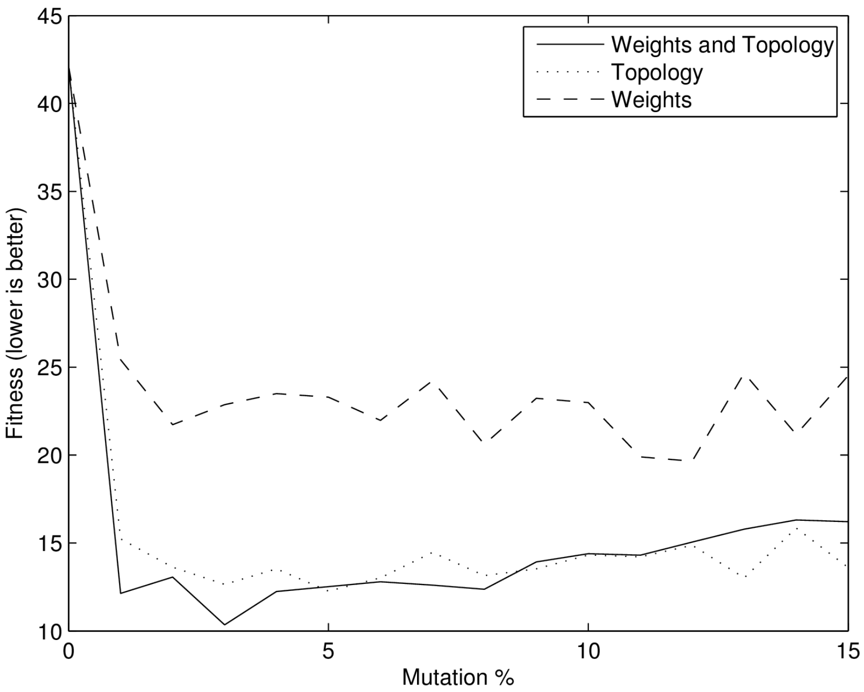
CGP Artificial Neural Networks (Evolved Topology)

Fitness is classification error i.e. lower is better

Evolving Topology



Double Pole Balancing
(Higher is better)

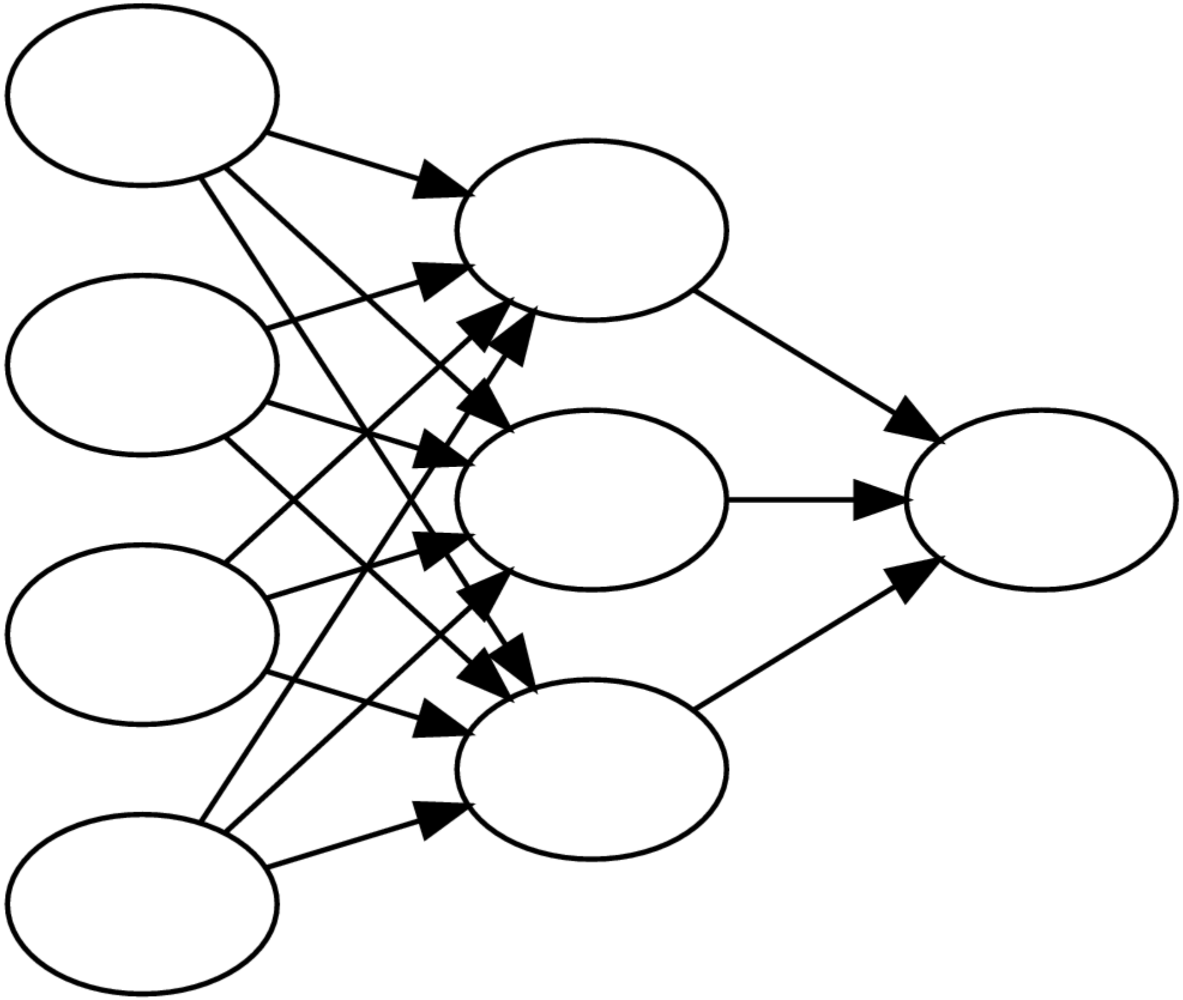


Monks Problem
(Lower is better)

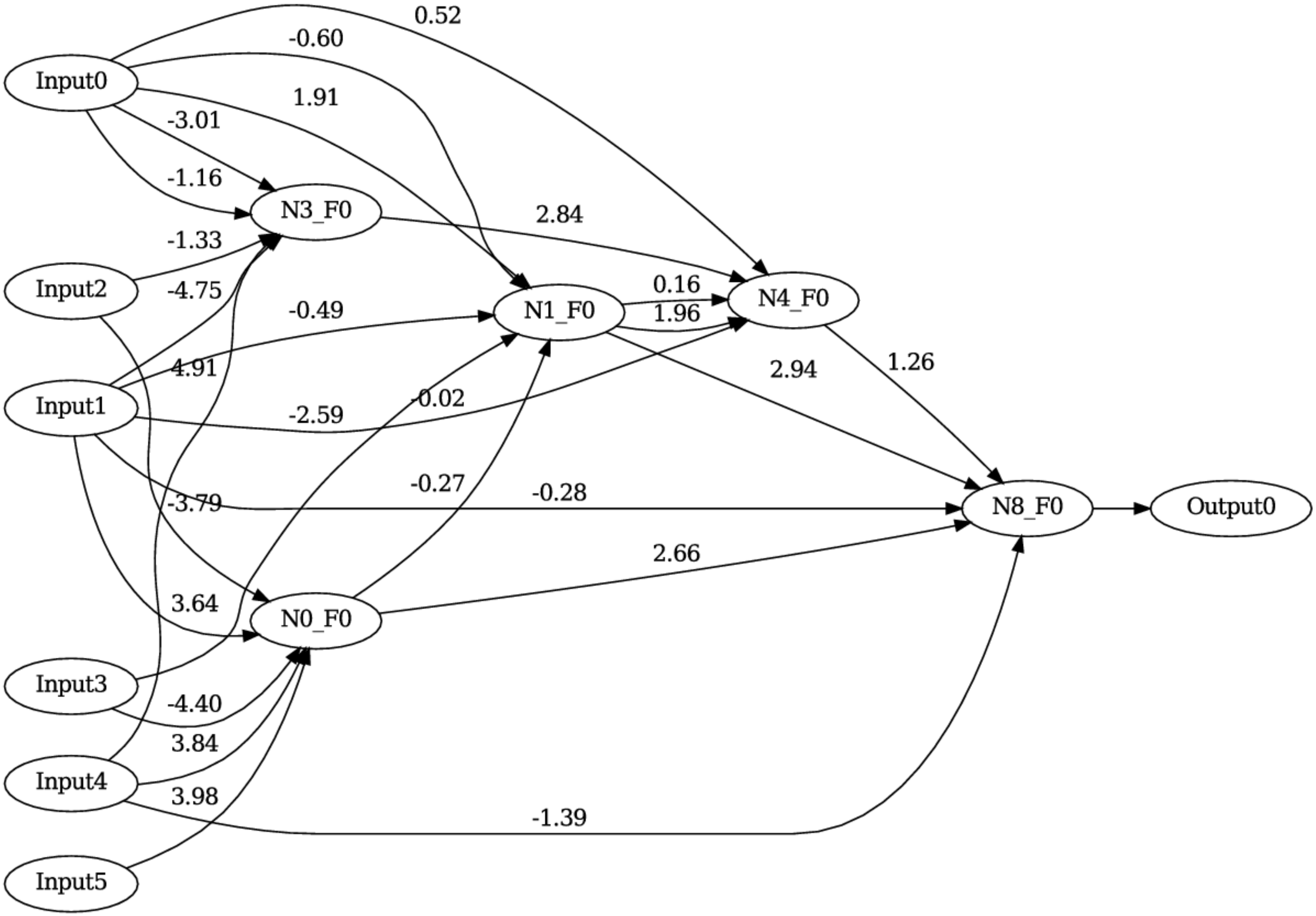
Evolving Topology

- So yes, topology drastically impacts on the effectiveness of NeuroEvolution.
- And evolution can be used to find suitable topologies.
- This is a **major advantage!** When using Conventional NeuroEvolution (or back propagation) one does not know which topologies will be suitable. Here evolution is finding suitable topologies for us.

Regular Neural Network



CGP Neural Network



Applications

- Applications to real tasks are important for machine learning research.
- They represent difficult tasks which demonstrate the capabilities of a given method.
- They are also used to compare different methods.
- And can also be quite fun :)

http://scr.geccocompetitions.com/results-of-the-2013-simulated-car-racing/

Questions?

CGPANN papers

- http://www.cartesiangp.co.uk/papers/gecco2013-turner.pdf
- http://www.sciencedirect.com/science/article/pii/S0925231213004499
- https://www.lri.fr/~hansen/proceedings/2012/GECCO/proceedings/p1031.pdf