CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

Neuroevolution

Alexandre Bergel
DCC - University of Chile
http://bergel.eu

 + 

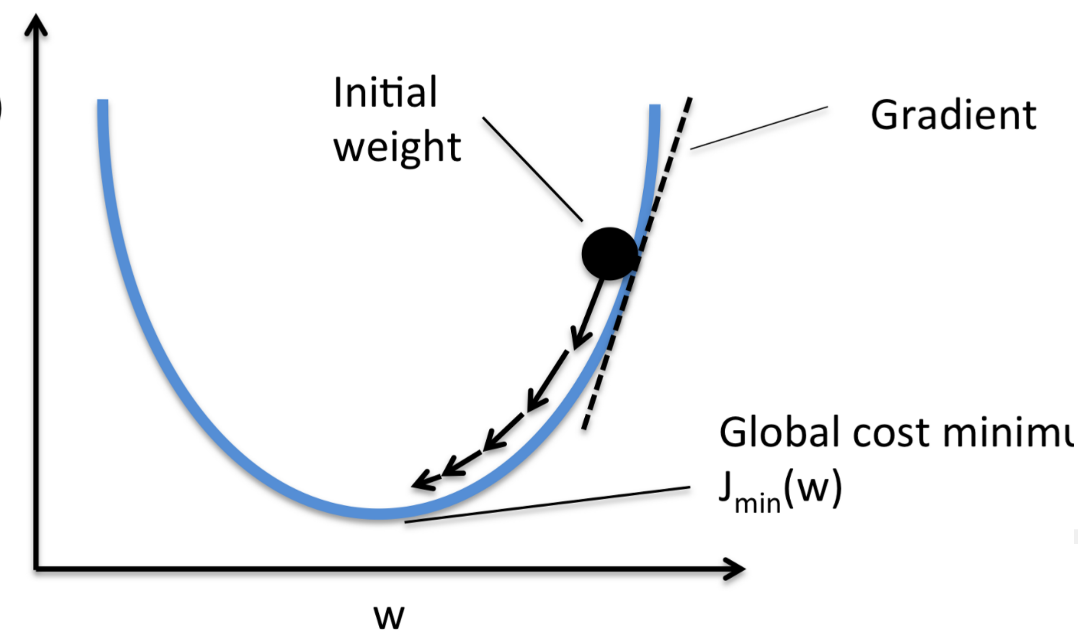
= Neuroevolution

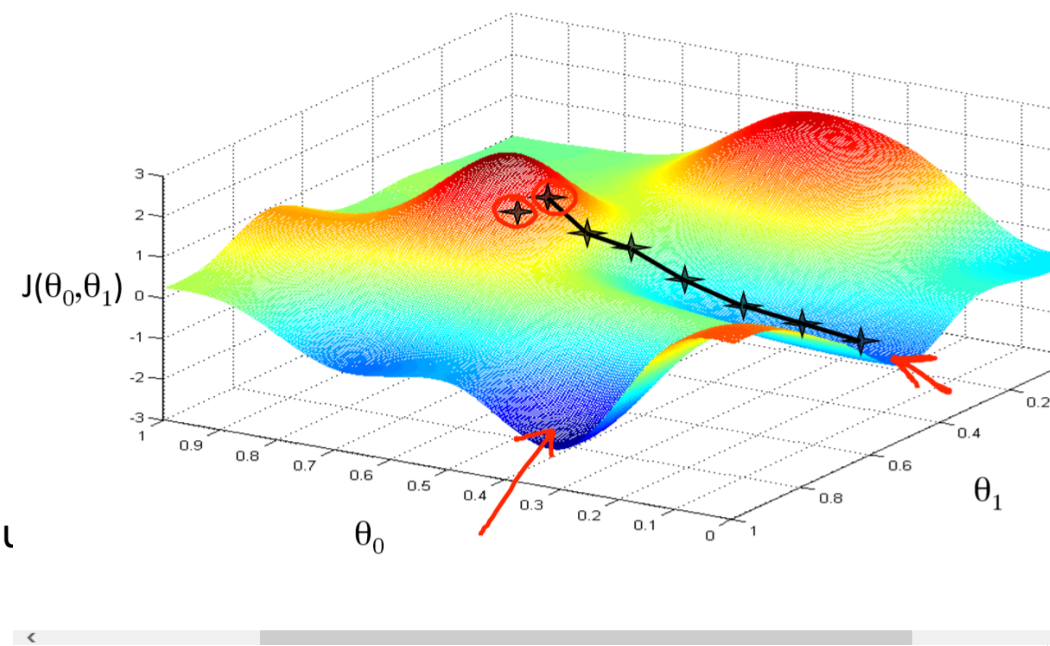
Goal of today

Overview of neuroevolution, a combination of *deep learning* and *genetic algorithm*

Gradient decent

Gradient descent is commonly used to make a neural network "learn"





Gradient descent requires examples

Examples are required to train a neural network

Picture recognition *may require thousands* of pictures

Requiring a large number of training examples may be *problematic*

e.g., video games, self-driving cars

Generalization

How do we get from where we are to *artificial general intelligence?*

How to make the learning process *closer to how human learn*?

https://www.nature.com/articles/s42256-018-0006-z

Neuroevolution

"Neuroevolution is a form of artificial intelligence that *uses evolutionary algorithms* to generate *artificial neural networks*, parameters, topology and rules. It is most commonly applied in artificial life, general game playing and evolutionary robotics."

— Wikipedia

"The evolution of neural networks is called neuroevolution"
— Hannah Le

Idea of Neuroevolution

Supervised learning algorithms require correct input-output pairs

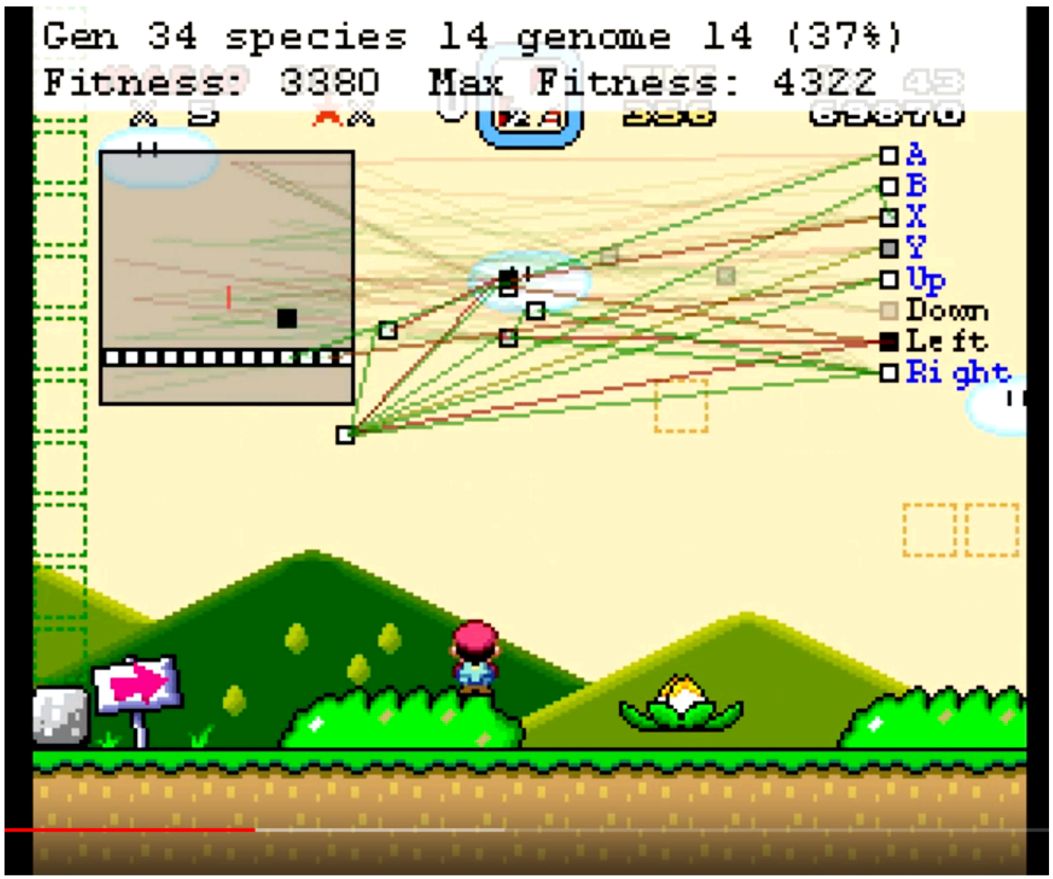
Neuroevolution only *requires a measure of a network's performance*

A major inspiration for the investigation of neuroevolution is the *evolution of brains* in nature

Examples of neuroevolution

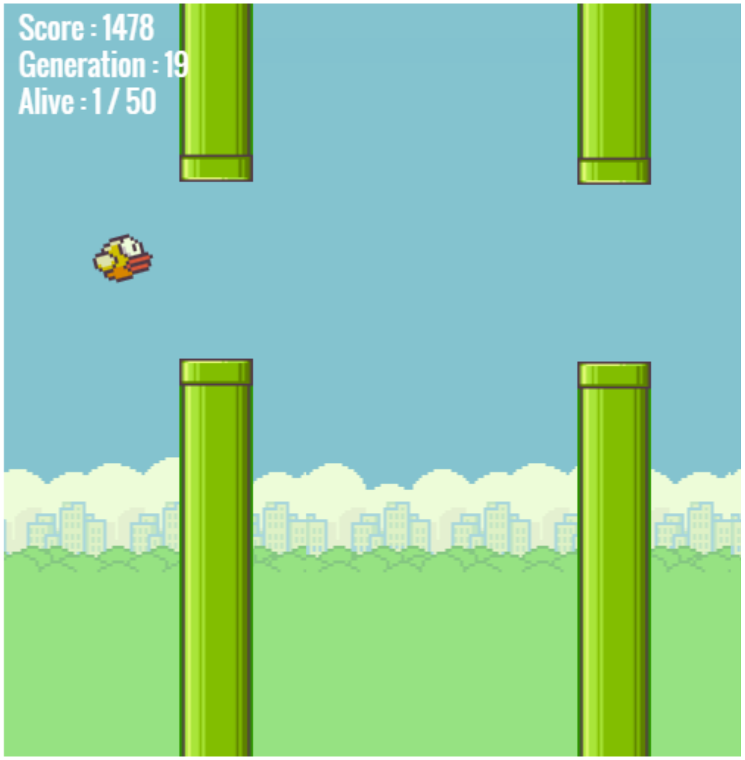
MarIO

https://www.youtube.com/watch?v=qv6UVOQ0F44



Flappy Bird

https://github.com/xviniette/FlappyLearning



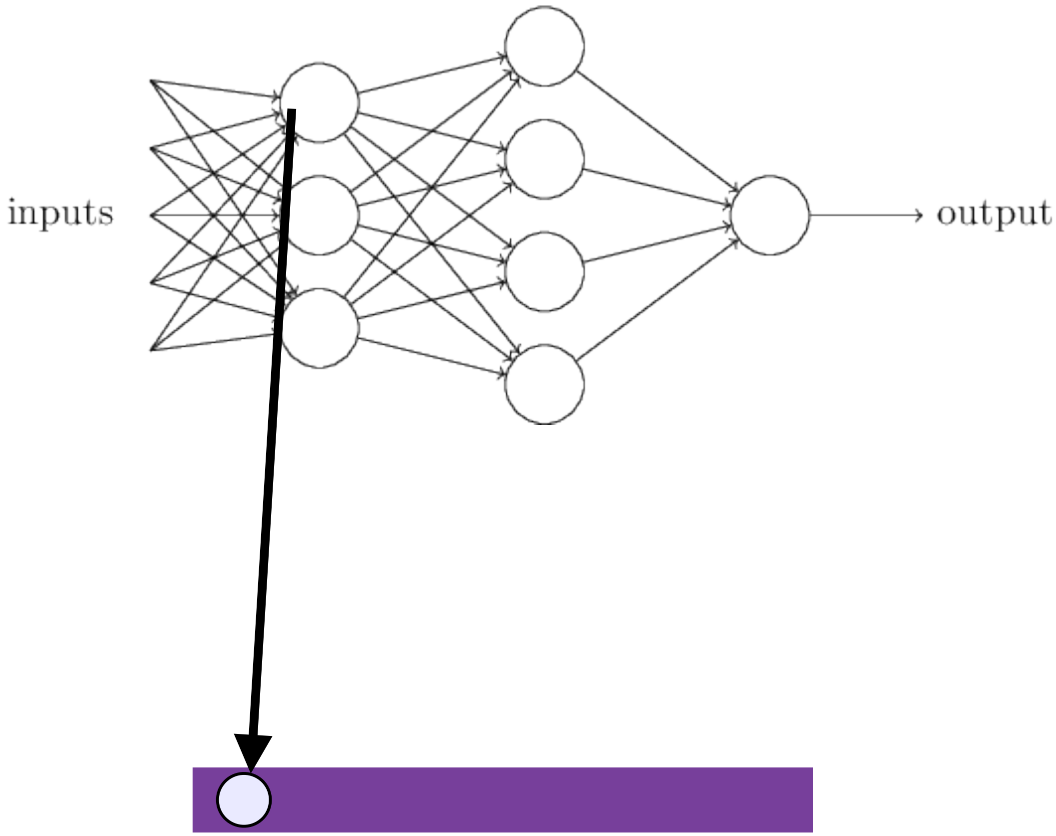
CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

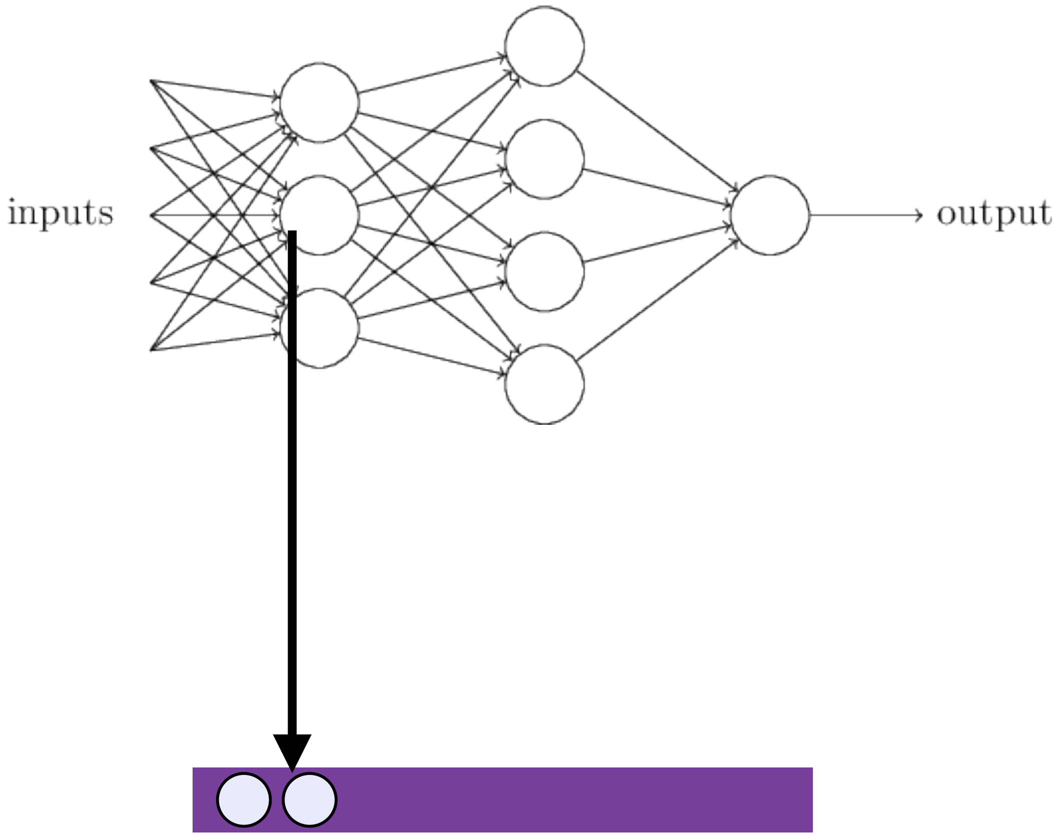
A simple algorithm for neuroevolution

Simple Algorithm

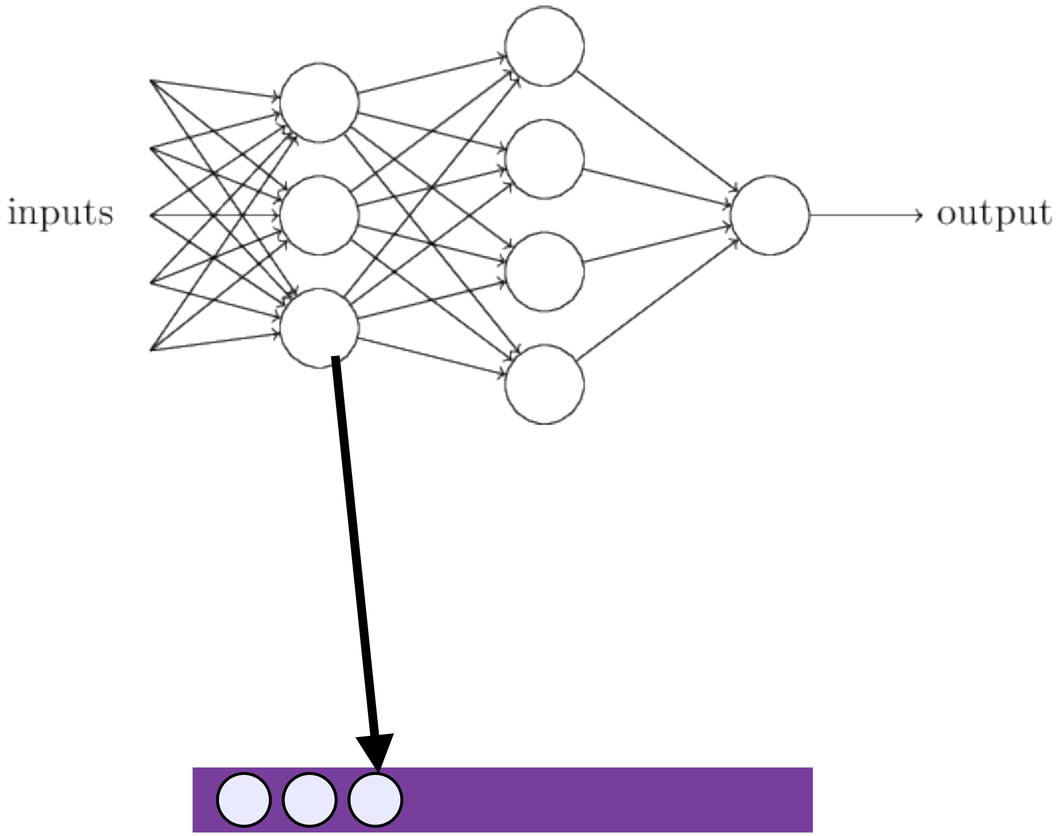
Use Genetic Algorithm to *search for the weights and bias* of each neuron

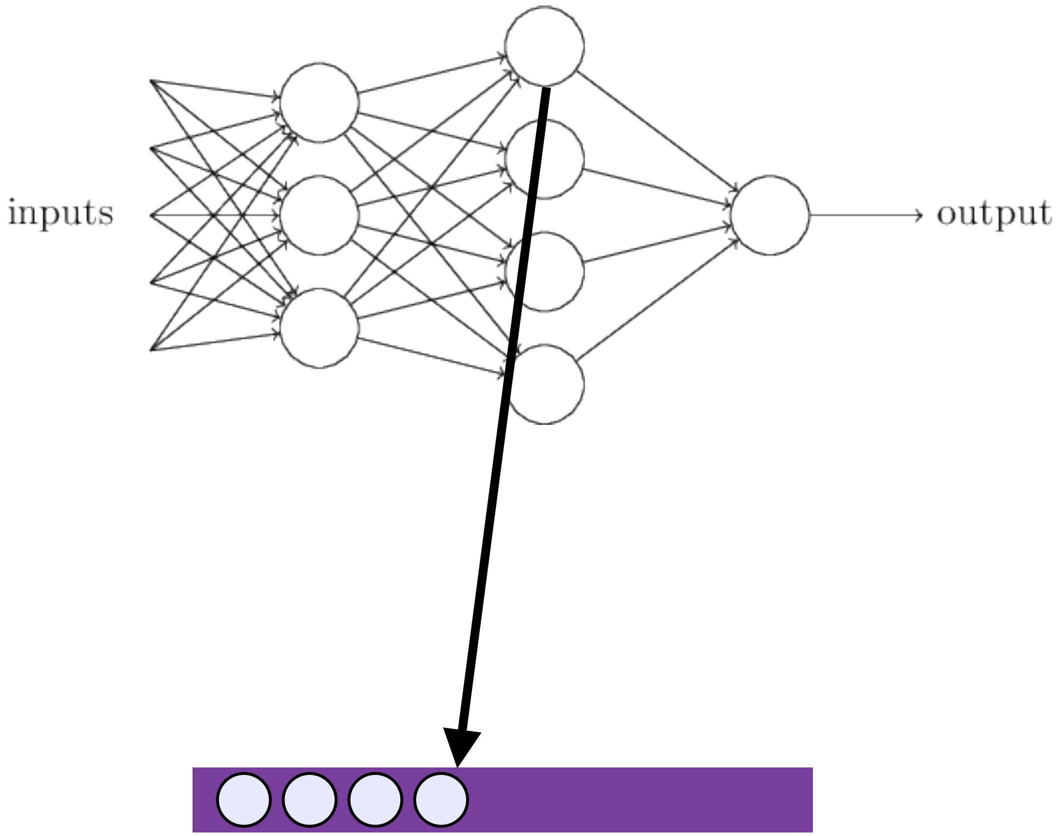
The fitness function to optimize could be based on the error function (e.g., MSE loss)

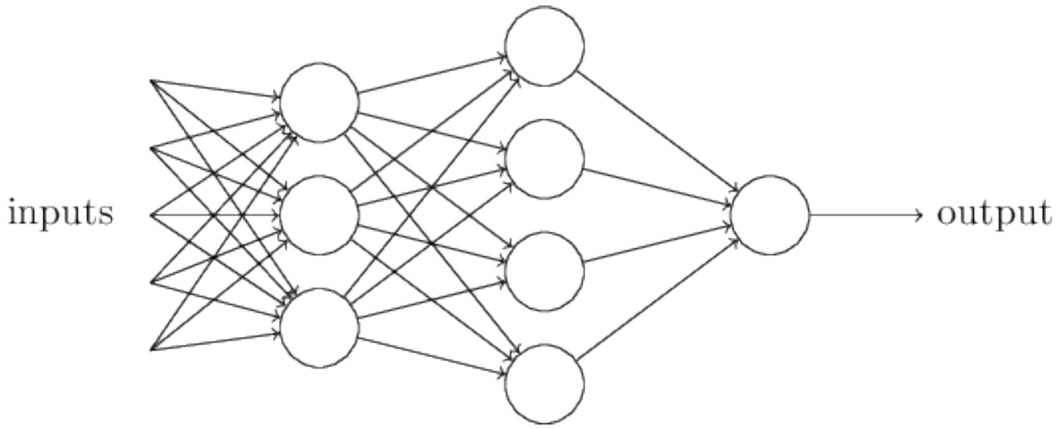

inputs
output

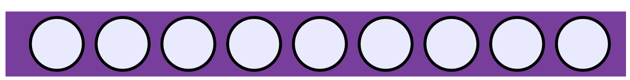

inputs
output

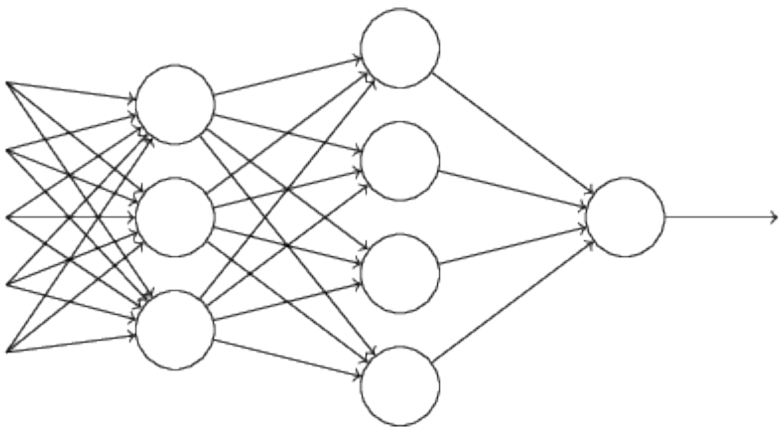
CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

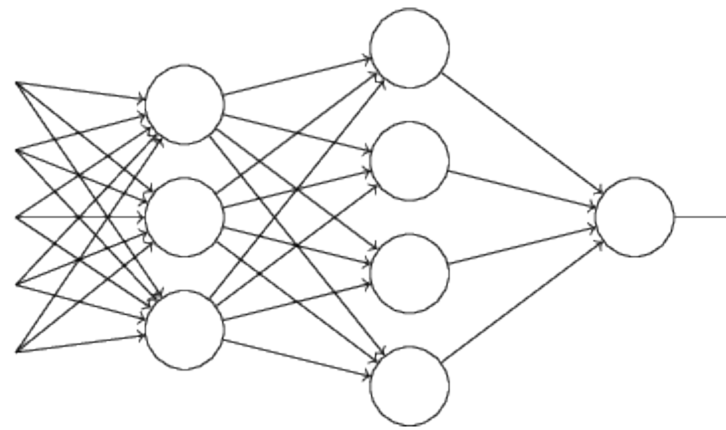


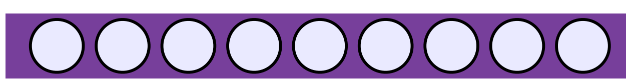

inputs
output

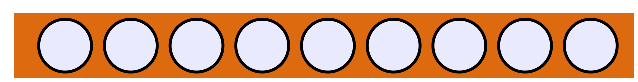

inputs
output

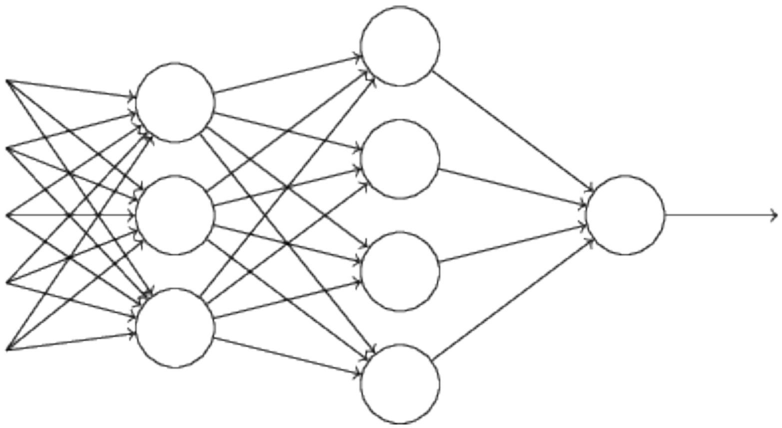


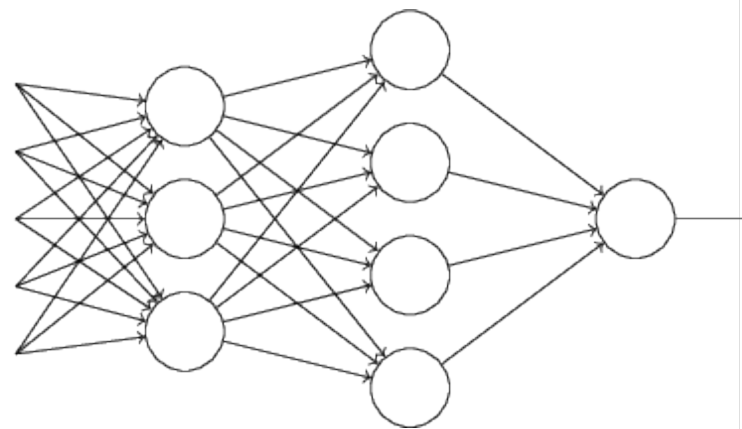


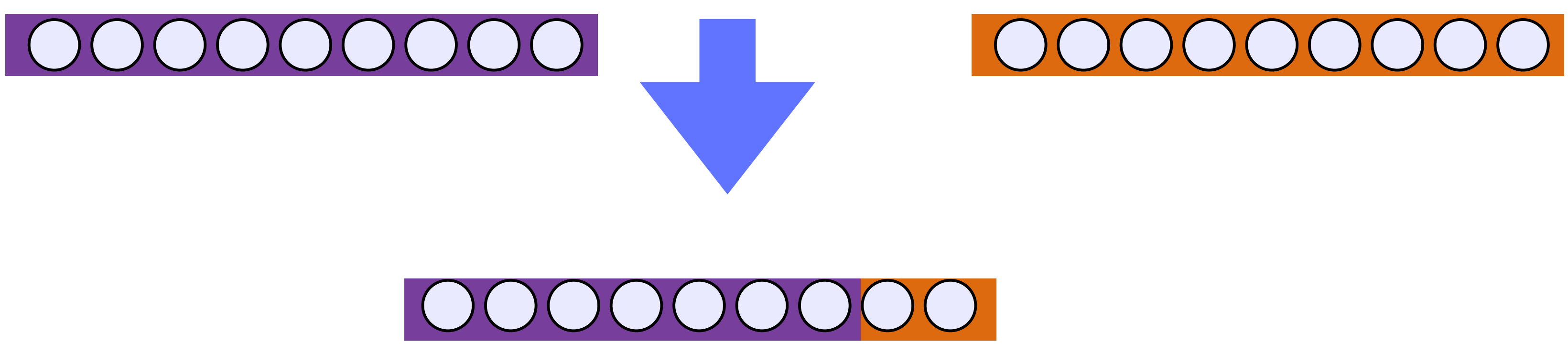


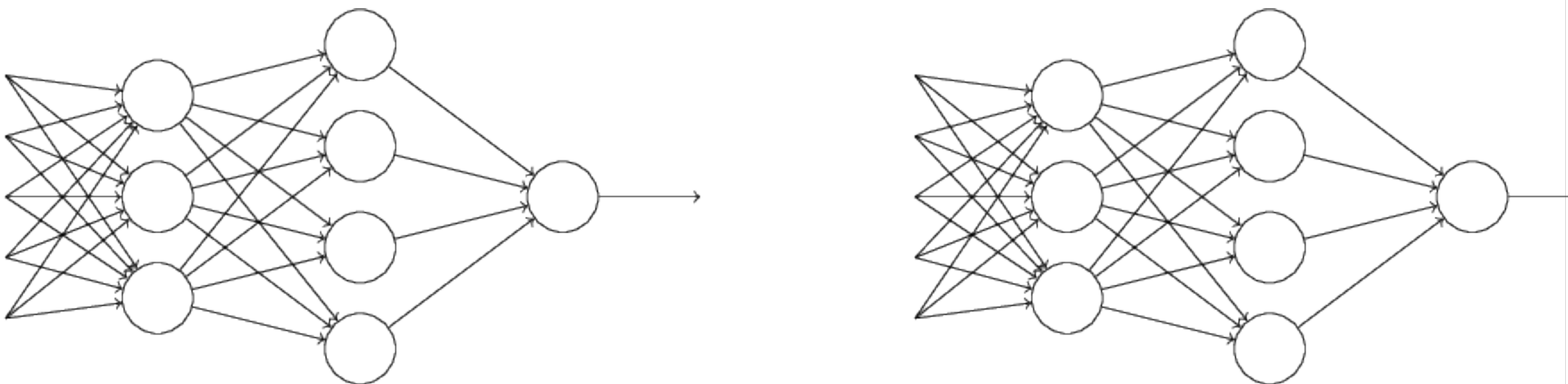




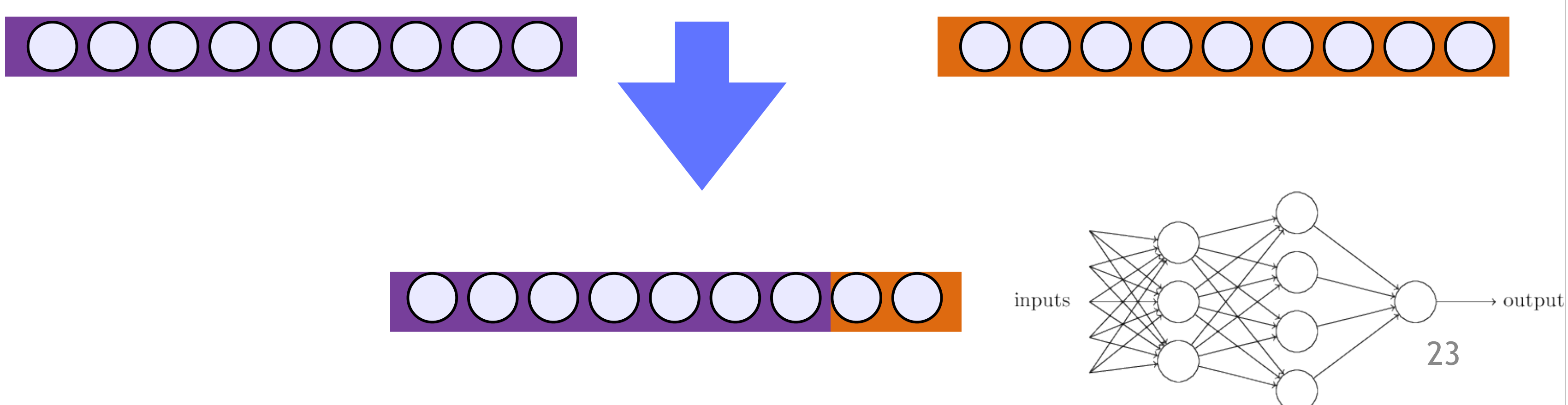








New neural network
With a better accuracy



Genetic operations

Applying genetic algorithm to find weights and bias of a neural network involves *particular genetic operations*

The cross-over operations *must consider the neuron* as a whole

ie. Crossover *should not divide the genetic information obtained from a neuron*

Cross-over happens only at a limit of a neuron

Because of the *competing convention* we will later see

Variant of crossover

We have originally presented GA with two genetic operations

In fact, we can run GA *with more operations*

You can also have an additional *second cross-over operations* that crosscut only at the limit of a layer

Cross-over happens at the *junction* of a layer

Genetically modified neural network

Neuroevolution is convenient when training data is either missing, insufficient, or unusable (eg bad quality, imbalanced)

The small algorithm we have just seen is *effective in many situations*

- Iris or seed data set
- AI for a simple game (e.g., Flappy bird)

What about network architecture?

This simple algorithm finds weights and bias while maintaining *a fixed network structure*

Structure of the network and some of its parameters can also be *discovered*

The idea of searching for a network architecture is to look for *simple and minimal networks*

Actually, fully-connected forward networks are pretty far away from what nature has brought us

NEAT — NeuroEvolution of Augmenting Topologies

Kenneth O. Stanley and Risto Miikkulainen. 2002.

Evolving neural networks through augmenting topologies

Evolutionary Computation archive

Volume 10 Issue 2, Summer 2002

Pages 99-127

http://dx.doi.org/10.1162/106365602320169811

Essence of NEAT

Genetic algorithm that *produces neural networks*

NEAT varies the *structure of the network*, producing complex solutions over time

NEAT needs its overall population of networks to be *diverse*

The reproduction phase should produces *a novel individual*

NEAT groups *similar individuals into species*

Direct competition for survival happens *only within a species*

Essence of NEAT

Species compete with each other for reproduction rights

Mating only happens within species (unless interspecies matin is explicitly allowed)

Because NEAT uses species, it *takes longer* for the whole population to become uniform

Limitation of gradient-based approaches

Neuroevolution can learn things that classical neural networks *cannot learn*

activation functions, hyperparameter, architectures

Encoding

Genome (Genotype)

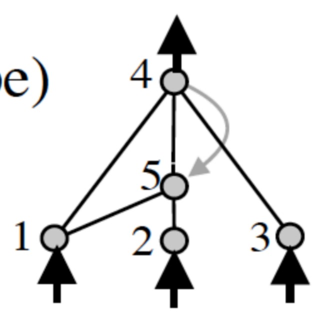
Node Genes

Node 1 Sensor	Node 2 Sensor	Node 3 Sensor	Node 4 Output	Node 5 Hidden

Connect. Genes

In 1	In 2	In 3	In 2	In 5	In 1	In 4
Out 4	Out 4	Out 4	Out 5	Out 4	Out 5	Out 5
Weight 0.7	Weight-0.5	Weight 0.5	Weight 0.2	Weight 0.4	Weight 0.6	Weight 0.6
Enabled	**DISABLED**	Enabled	Enabled	Enabled	Enabled	Enabled
Innov 1	Innov 2	Innov 3	Innov 4	Innov 5	Innov 6	Innov 11

Network (Phenotype)



Genome (Genotype)

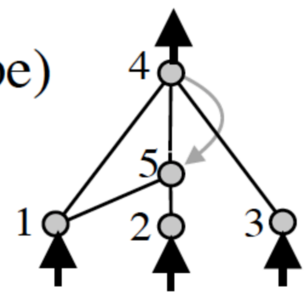
Node Genes

Node 1 Sensor	Node 2 Sensor	Node 3 Sensor	Node 4 Output	Node 5 Hidden

Connect. Genes

In 1 Out 4 Weight 0.7 Enabled Innov 1	In 2 Out 4 Weight-0.5 **DISABLED** Innov 2	In 3 Out 4 Weight 0.5 Enabled Innov 3	In 2 Out 5 Weight 0.2 Enabled Innov 4	In 5 Out 4 Weight 0.4 Enabled Innov 5	In 1 Out 5 Weight 0.6 Enabled Innov 6	In 4 Out 5 Weight 0.6 Enabled Innov 11

Network (Phenotype)



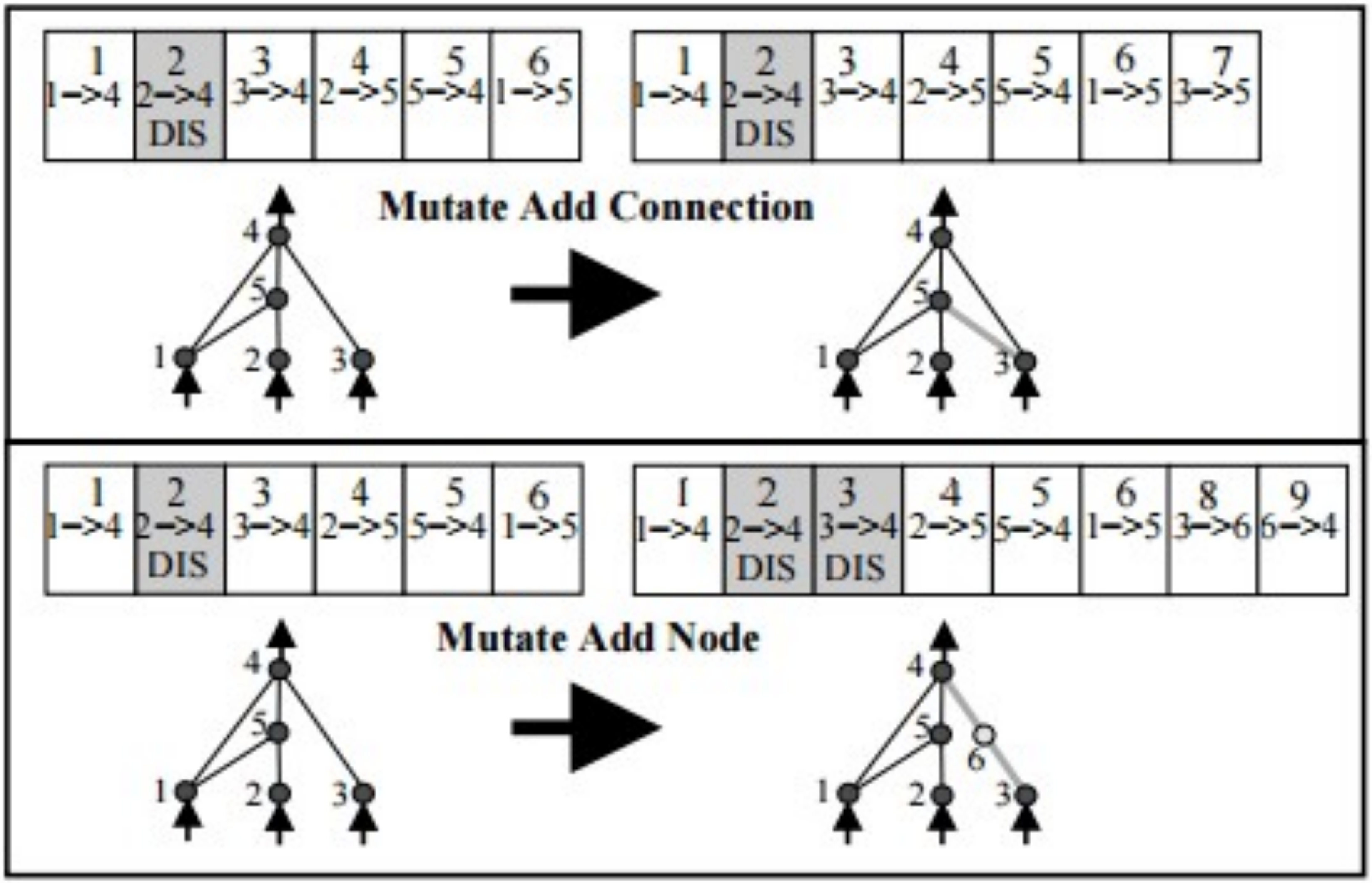
Inputs and outputs nodes are not encoded

Hidden nodes can be added or removed

A connection may be enabled or not, and has an innovation number

Individuals do not necessarily have the same size

Mutation


1 1->4
2 2->4 DIS
3 3->4
4 2->5
5 5->4
6 1->5
1 1->4
2 2->4 DIS
3 3->4
4 2->5
5 5->4
6 1->5
7 3->5
Mutate Add Connection
1 1->4
2 2->4 DIS
3 3->4
4 2->5
5 5->4
6 1->5
1 1->4
2 2->4 DIS
3 3->4 DIS
4 2->5
5 5->4
6 1->5
8 3->6
9 6->4
Mutate Add Node

Mutation

If a new node is added:

1 - it is placed between two connected nodes

2 - the previous connection is disabled (but still present in the genome)

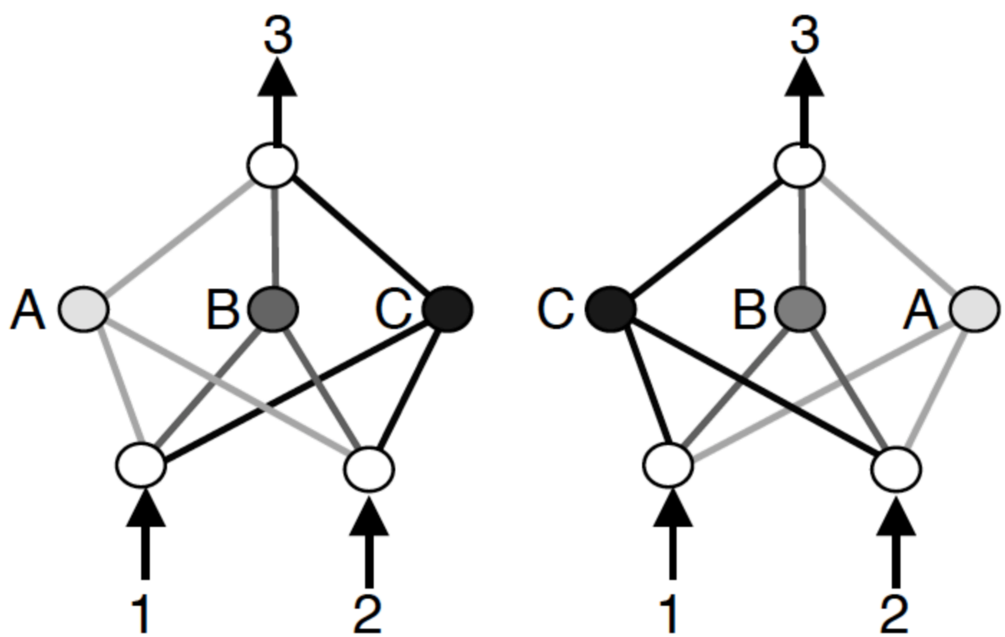
Competing Convention Problem

In ordinary evolution algorithms it can happen that two individuals encode very *similar behavior*, but with very *different genotype*

This is called *competing conventions*

If these individual are subject to crossover, their children are likely to be *worse* than their either parent

Competing Convention Problem



[A,B,C]
X[C,B,A]

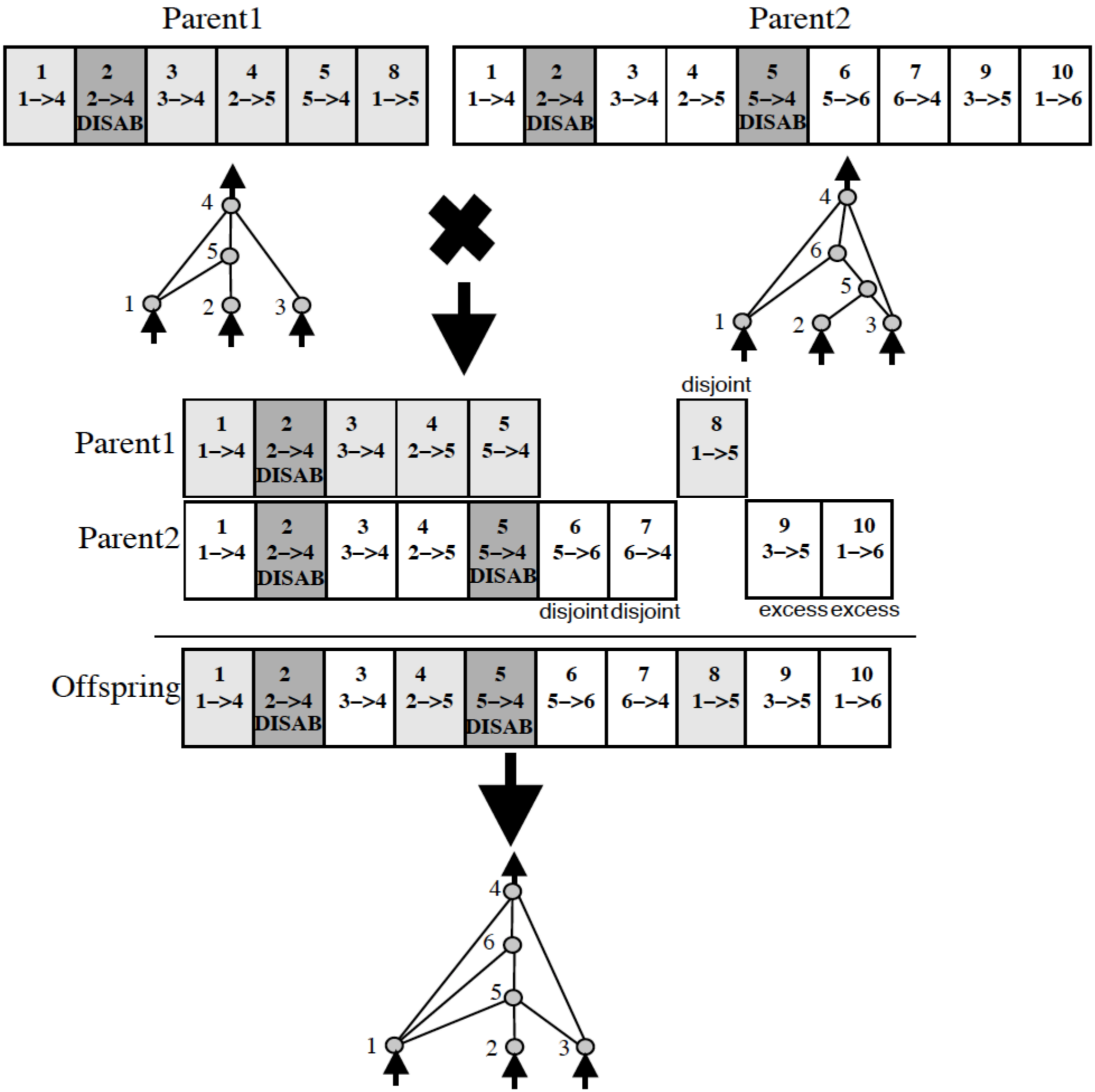
Crossovers: [A,B,A] [C,B,C]
(both are missing information)

Competing Convention Problem

NEAT solves this by *keeping historical markings* of new structural elements

When a new structural element is created (via structural mutation, such as adding a new node or gene), it receives an *innovation number*

When two individuals are crossed over, their genotypes are *aligned* in such a way that the corresponding innovation numbers match and only the different elements are exchanged


Parent1
1 1->4
2 2->4 DISAB
3 3->4
4 2->5
5 5->4
8 1->5
Parent2
1 1->4
2 2->4 DISAB
3 3->4
4 2->5
5 5->4 DISAB
6 5->6
7 6->4
9 3->5
10 1->6
disjoint
disjoint
disjoint
excess
excess
Offspring
1 1->4
2 2->4 DISAB
3 3->4
4 2->5
5 5->4 DISAB
6 5->6
7 6->4
8 1->5
9 3->5
10 1->6

Speciation

Most new evolutions *are not good ones*

A new connection or node degrades the performance of the individual

How can we protect new networks from being eliminated?

NEAT suggests *speciation*

Speciation

Splits up the *population into several species*

Based on the *similarity of topology and connections*

Individuals in a population only have to *compete* within other individuals *within that species*

Allow for a *new network to be created and optimized* without being quickly eliminated

Explore the idea of *explicit fitness sharing*: the whole population has a fitness

Speciation

Speciation *project innovations in the population*

When a network diverges far enough from other networks in the population, NEAT identifies it and put it in its own species

Using innovation numbers NEAT *computes the distance between two genomes*

Minimal structure

NEAT favors *minimal networks to be evolved*

Mutations is about adding a node, adding a connection, or changing a weight

… and not removing structural elements

What have we seen?

Overview of what Neuroevolution is

Two algorithms: simple and NEAT

We want more!

Many improvements were proposed HyperNEAT, ESHyperNEAT, CoDeepNEAT

Indirect coding using rules mimicking cell divisions dcc

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

www.dcc.uchile.cl

/ DCCUCHILE