

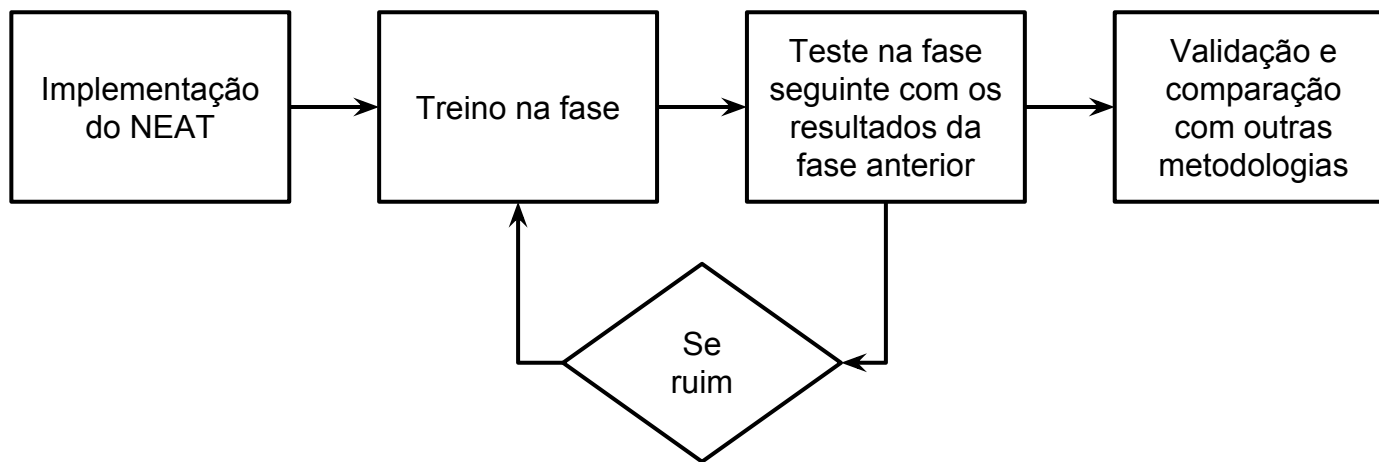
Visão Computacional

Usando Neuro Evolução com Crescimento Topológico para
jogar Mário

O Problema e por que

- É possível modelar um sistema para jogar Mario?
- Mario é um jogo mais complexo que possui uma variedade de situações novas a cada fase.
- Usaremos o NEAT por ele modelar melhor o comportamento evolucionário da natureza e do cérebro.
- O NEAT possui uma melhor performance em relação à outras técnicas de Neuroevolução.

Metodologia



NEAT

- *NeuroEvolution of Augmenting Topologies* (NEAT) é um algoritmo genético para gerar redes neurais artificiais.
- Altera tanto os pesos quanto a estrutura geral da rede.
- Baseia-se em 3 técnicas:
 - *Tracking* dos genes com marcadores históricos para permitir *crossover* entre topologias.
 - Aplicação de *speciation* para preservar inovação.
 - Desenvolver topologias incrementalmente a partir de estruturas simples.

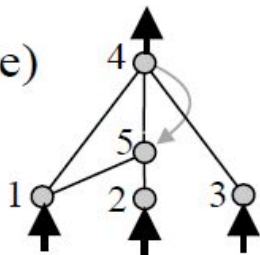
Diferença entre NEAT e outros Algoritmos Genéticos

- Minimização da dimensionalidade.
- Faz o *tracking* da origem dos genes.
- Proteção de inovação através de *speciation*.
- Aplicação de um método de *crossover* entre diferentes *topologias*.

Estrutura

Genome (Genotype)						
Node	Node 1	Node 2	Node 3	Node 4	Node 5	
Genes	Sensor	Sensor	Sensor	Output	Hidden	
Connect. Genes	In 1	In 2	In 3	In 2	In 5	In 1
	Out 4	Out 4	Out 4	Out 5	Out 4	Out 5
	Weight 0.7	Weight -0.5	Weight 0.5	Weight 0.2	Weight 0.4	Weight 0.6
	Enabled	DISABLED	Enabled	Enabled	Enabled	Enabled
	Innov 1	Innov 2	Innov 3	Innov 4	Innov 5	Innov 6
					In 4	Out 5
						Weight 0.6
						Enabled
						Innov 11

Network (Phenotype)



Minimização de dimensionalidade

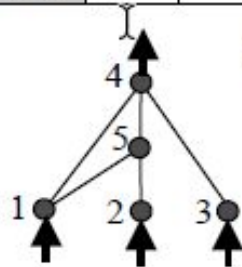
- O NEAT começa sem camadas ocultas e todas as entradas são diretamente ligadas às saídas a partir das quais ele vai evoluindo o valor dos pesos e criando novas topologias.
- Começando sem camadas ocultas, e através das 3 técnicas garantido que somente alterações benéficas vão sendo adicionadas à estrutura e que não existirão partes desnecessárias, assim tendendo à minimização da dimensionalidade da solução.

Historical Matching

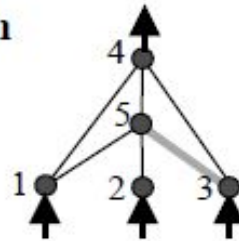
- Dois genes que possuem o mesmo ancestral em comum, possuem a mesma estrutura, apesar de possuir pesos diferentes.
- A técnica de *Historical Matching* consiste em marcar os genes que possuem um mesmo ancestral em comum.
- Utilizando um contador global chamado de número de inovação (*global innovation number*) cada gene recebe um número, que é incrementado à medida que novos genes vão surgindo.
- No *crossover*, genes com o mesmo número de inovação são alinhados, e genes com número de inovação diferentes são selecionados do progenitor com melhor *fitness* para serem passados ao filho.

1	2	3	4	5	6
1->4	2->4 DIS	3->4	2->5	5->4	1->5

1	2	3	4	5	6	7
1->4	2->4 DIS	3->4	2->5	5->4	1->5	3->5

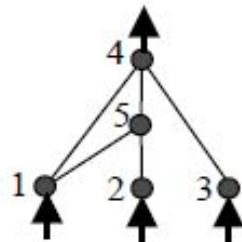


Mutate Add Connection

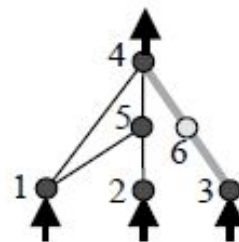
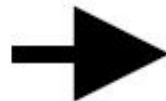


1	2	3	4	5	6
1->4	2->4 DIS	3->4	2->5	5->4	1->5

1	2	3	4	5	6	8	9
1->4	2->4 DIS	3->4 DIS	2->5	5->4	1->5	3->6	6->4



Mutate Add Node

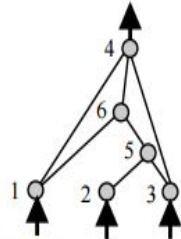
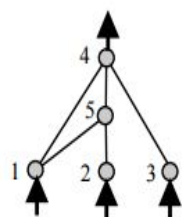


Parent1

Parent2

1	2	3	4	5	8
1->4	2->4 DISAB	3->4	2->5	5->4	1->5

1	2	3	4	5	6	7	9	10
1->4	2->4 DISAB	3->4	2->5	5->4 DISAB	5->6	6->4	3->5	1->6



disjoint

Parent1

1	2	3	4	5
1->4	2->4 DISAB	3->4	2->5	5->4

8
1->5

Parent2

1	2	3	4	5	6	7
1->4	2->4 DISAB	3->4	2->5	5->4 DISAB	5->6	6->4

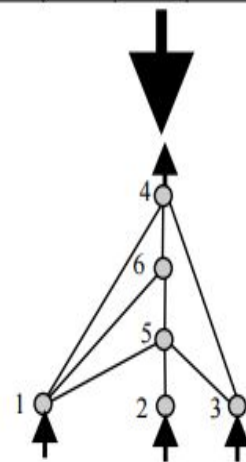
disjoint disjoint

9	10
3->5	1->6

excess excess

Offspring

1	2	3	4	5	6	7	8	9	10
1->4	2->4 DISAB	3->4	2->5	5->4 DISAB	5->6	6->4	1->5	3->5	1->6



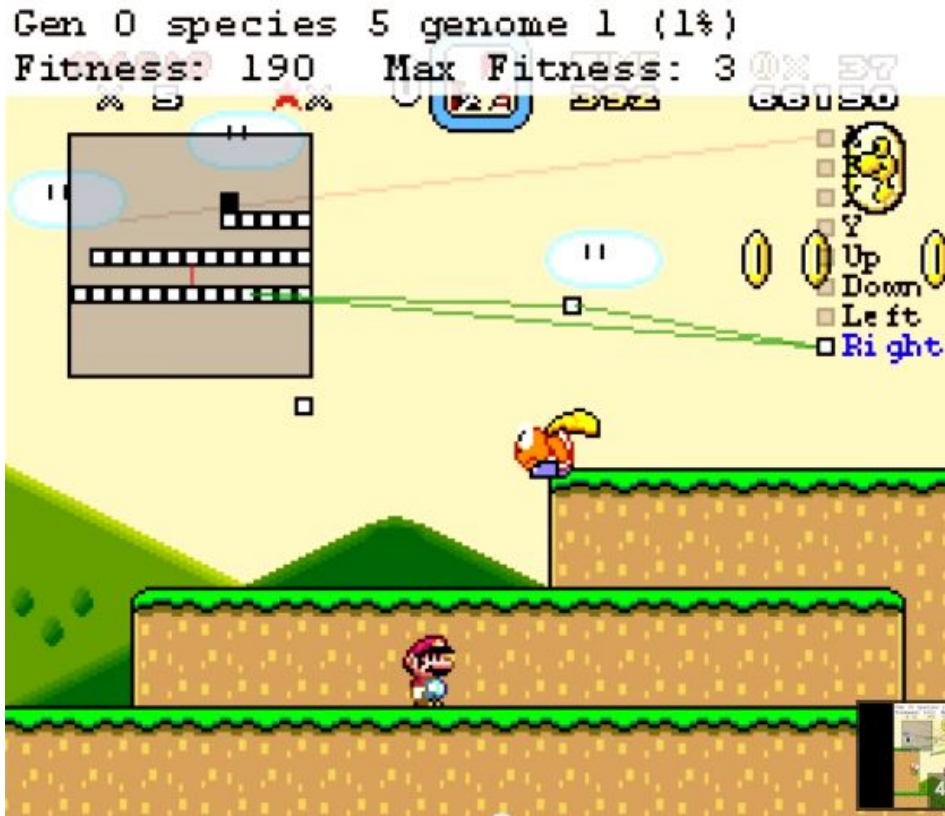
Preservação de inovações

- Adicionar um novo nó à estrutura geralmente afeta negativamente o fitness, o que pode fazer com que uma estrutura seja precocemente descartada.
- Para resolver isso é usado *speciation*, permitindo que as novas estruturas compitam inicialmente em um nicho específico da mesma espécie em vez da população global, permitindo assim que essas inovações evoluam antes de competir com a população global.

NEAT para jogar Mario

- Cada elemento da fase (terreno e inimigos) representa uma entrada na rede.
- As saídas são os botões que a rede irá apertar (A, B, X, Y, Up, Down, Left, Right).
- Uma entrada pode não estar ligada a uma saída e vice-versa.
- A partir das leituras do emulador as entradas são fornecidas e as saídas são geradas.

NEAT para jogar Mario



Referências

1. Stanley, Kenneth O., and Risto Miikkulainen. "Evolving neural networks through augmenting topologies." *Evolutionary computation* 10.2 (2002): 99-127.
2. Monteiro, Ivan Medeiros. "Neuroevolução com Crescimento Topológico."
3. Bizhawk, <https://github.com/TASVideos/BizHawk>
4. <https://www.engadget.com/2015/06/17/super-mario-world-self-learning-ai/>