CIG 2007 Tutorial

Practical Issues in Evolving Neural Network Controllers for Video Game Agents

Kenneth O. Stanley

The University of Central Florida

April, 2007

E P L E X

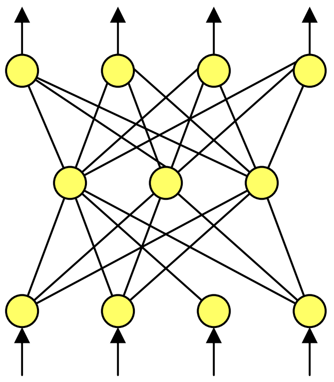
Evolutionary Complexity Research Group at UCF

http://eplex.cs.ucf.edu

Why Machine Learning in Video Games?

- Better player experience
  - Agents can adapt to player
  - Increased variety of agent behaviors
  - Ever-changing behaviors
  - – Entirely new game genres possible
- What is the goal?
  - – Fun!
  - – Not “optimal performance”

Why Artificial Neural Networks (ANNS) In Video Games?



- Continuous control
- Resistant to noise
- Low CPU burden
- Many domain-general tweaks
  - Sensitive/weaken sensors
  - Strengthen/weaken outputs
  - Slightly perturb weights
  - Add/substract inputs/outputs
- Evolvable!

Should ANNs Be Evolved In the Game or Before the Game?

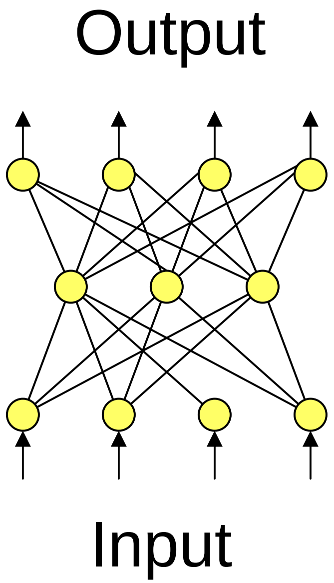
- Depends what you’re trying to do
  - Find a good behavior: Before
  - Prevent player exploitation: In-Game
- *In-Game* is a lot harder
  - But leads to interesting new possibilities

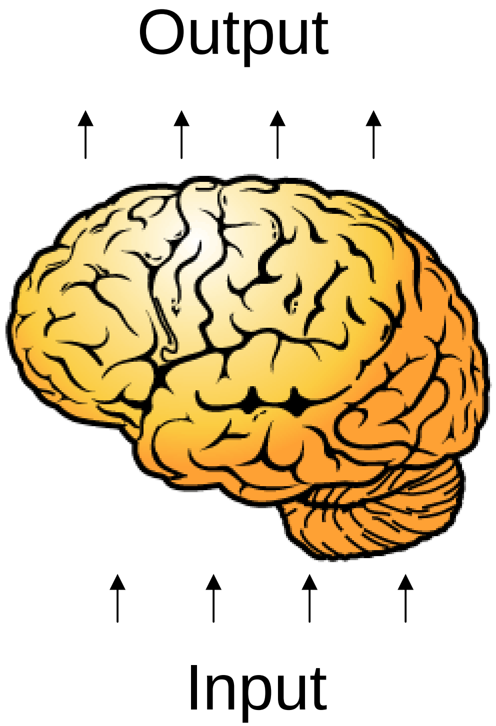
Why Evolve ANNs In Video Games?

- Evolution is a simple answer to the reinforcement learning / credit assignment problem for video games
  - Action selection instead of value estimation works well in high-dimensional state/action space
  - A population includes *diverse* behaviors
  - Consistent individual behaviors
  - Fast adaptation of small networks
  - Memory of past states

Basic Concepts

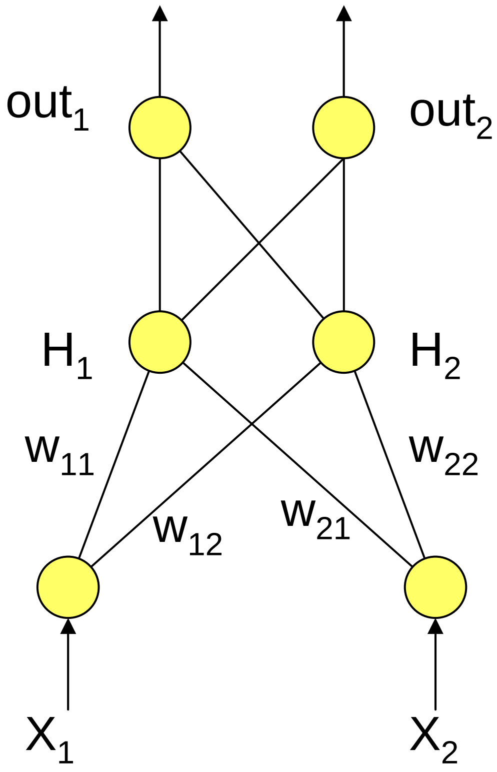
- Artificial Neural Networks (ANNs)
  - Loose abstraction of brains
  - Can be used to control NPCs





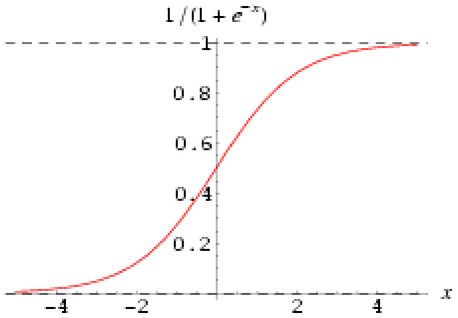
Basic Concepts

- ANN activation



Neuron *j* activation:

$$H_j = \sigma\left(\sum_{i=1}^{n} x_i w_{ij}\right)$$



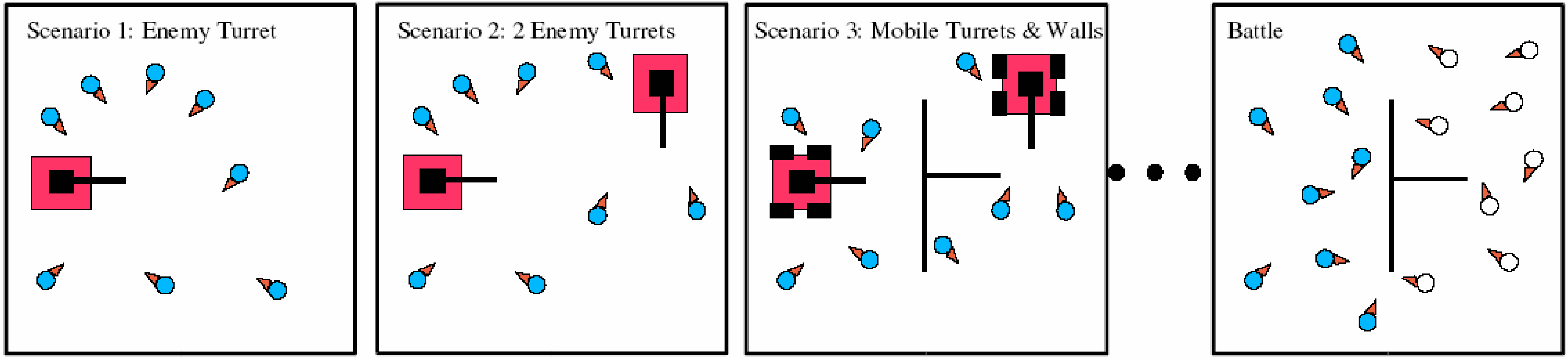
Basic Concepts

- Evolutionary Computation (Genetic Algorithms)
  - Create a population of candidates
  - Loop
    - Evaluate the entire population the task (fitness)
    - Choose the better individuals
    - Mate (crossover) and mutate to form next generation
    - Repeat loop
- Darwinian survival-of-the-fittest takes hold

Basic Concepts

- ANNs + Evolution = Neuroevolution
- If NPCs are controlled by ANNs, then neuroevolution can happen *inside* the game
- NERO is an example

NERO: NeuroEvolving Robotic Operatives



- *Training Game*: Train your troops for combat
- Player trains NPCs for goal and style of play
- NPCs improve in real time as game is played

NERO Battle Mode

- After training, learned behaviors are saved
- Player assembles team of trained agents
- Team is tested in battle against opponent’s team

The NERO Story

- Over 30 volunteers over 2 years
- Free download at http://nerogame.org
- Playable demos have public appeal
  - Over 80,000 downloads
  - Appeared on Slashdot
  - Best Paper Award in Computational Intelligence and Games (IEEE CIG 05)
  - Independent Games Festival Best Student Game Award (GDC 2006)
  - Worldwide media coverage

Media Coverage

Media Coverage

FIZMATI.LV
Latvijas Universitātes Fizikas un matemātikas fakultātes vortāls

Sākums | Ziņas | Dzīve | Studijas | Resursi | Forums

Izvēlne

Ziņas
- Datorika
- Padomi
- Zinātne
- Notikumi
- Citas tēmas
- Iesūtīt rakstu

Dzīve
Studijas
Resursi
Forums

Meklēšana

Meklēt

Aptauja

Kā pavadīsi vasaru?

strādāšu

Ziņas / Datorika / Studenti rada jauna tipa spēli

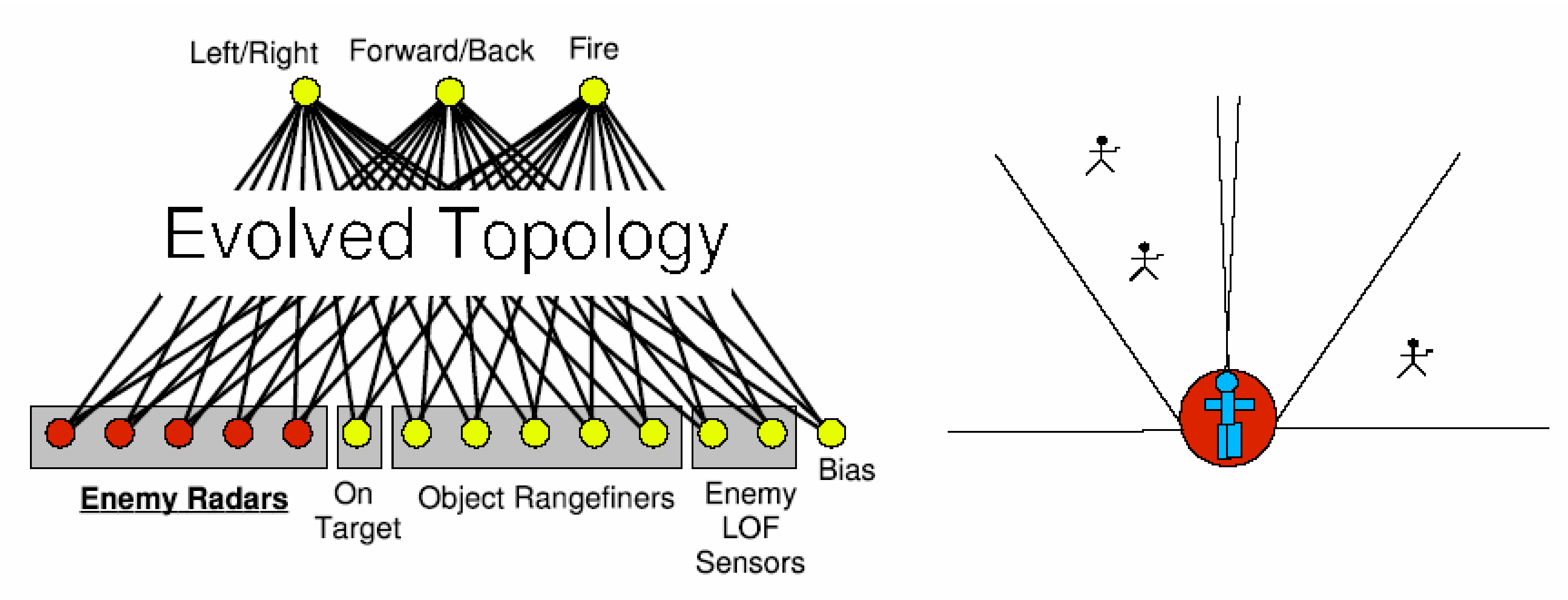
Studenti rada jauna tipa spēli (2)

2005. gada 29. jūnijā 17:43

Aptuveni 30 studentu liela grupa no Teksasas Universitātes Datorzinātņu nodaļas Neironu tīklu izpētes grupas ir radījuši jaunu spēli tādējādi nopietni aizsākot arī jaunu spēļu žanru. Viņu izstrādātā spēle "NERO" izmanto akadēmiskus mākslīgā intelekta pētījumu rezultātus un ļauj spēlētājiem apmācīt savu virtuālu robo-kareivju vienību. Spēlētājs var iemācīt saviem mākslīga intelekta vadītajiem padotajiem dažādas taktiskas viltības.

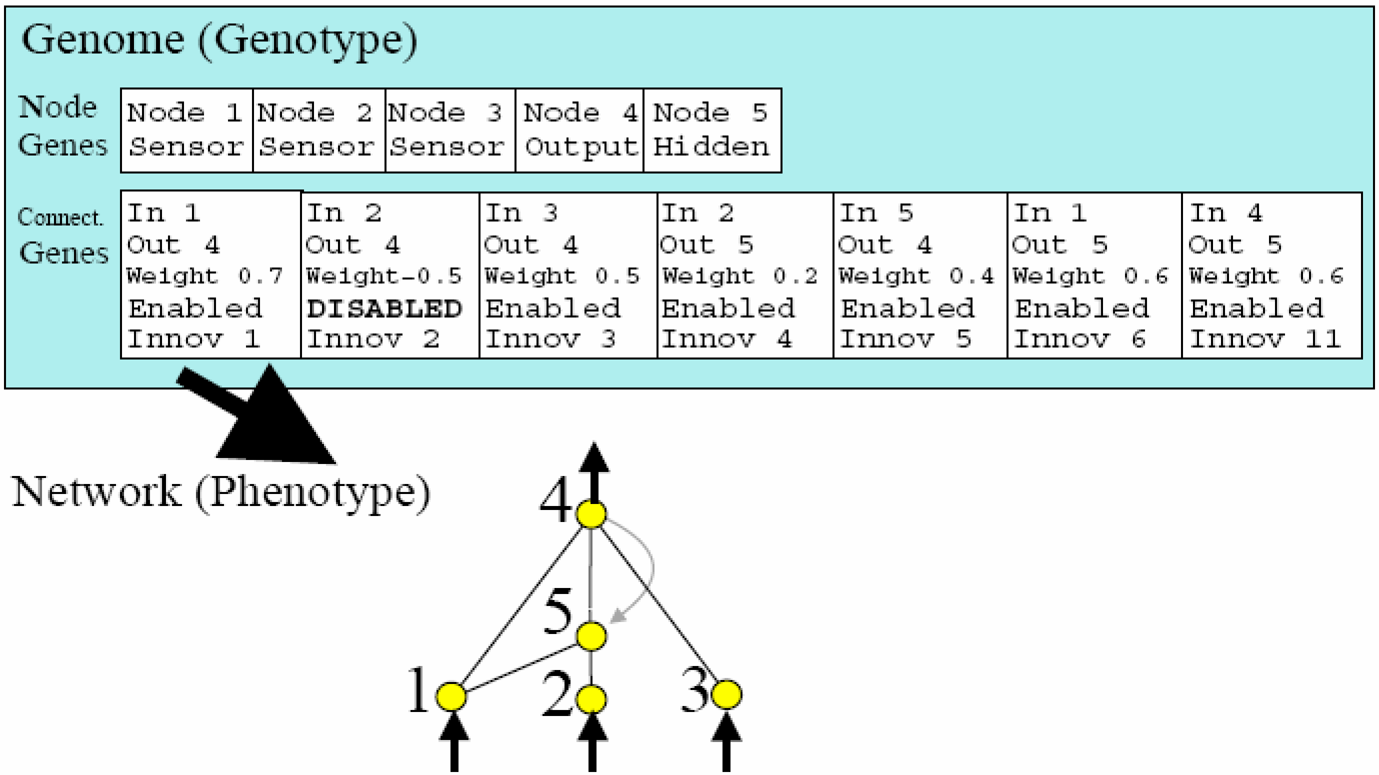
"NERO" (Neuro-Evolving Robotic Operatives) aizsāktais spēļu žanrs ir iespējams tikai pateicoties mašīnu apmācības tehnoloģijai. Lai arī "NERO" atgādina parastu RTS, tā nav, jo atšķirībā no parastām RTS šai spēlei ir divas izteiktas spēles fāzes. Pirmajā fāzē spēlētāji pilda trenera un skolotāja lomu apmācot nepieredzējušos kareivjus dažādām kaujas mākslām. Kad šie kareivji ir pietiekami labi apmācīti, spēlētāji pārslēdzas uz otro fāzi, kur viņu audzēkņu iemaņas tiek pārbaudītas cīņās ar citu spēlētāju apmācītajiem kareivjiem.

Learning Method: Real-time NeuroEvolution of Augmenting Topologies (rtNEAT)

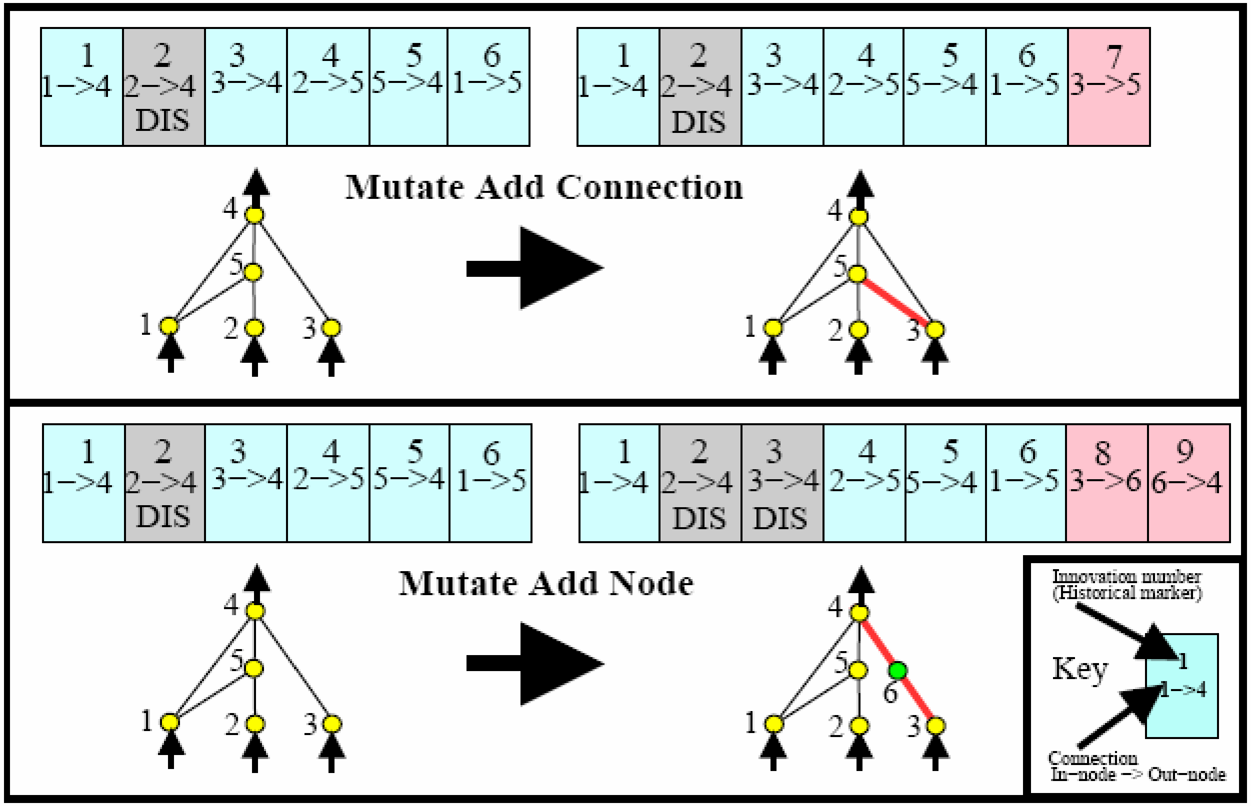


- Allows increasingly complex ANNs to evolve in real time as the game is played

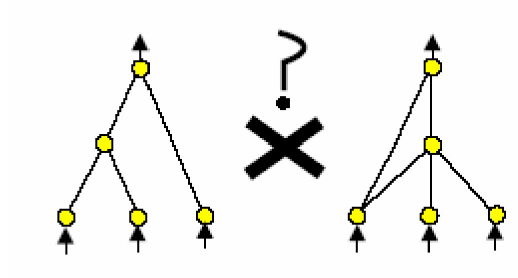
Genetic Encoding in NEAT



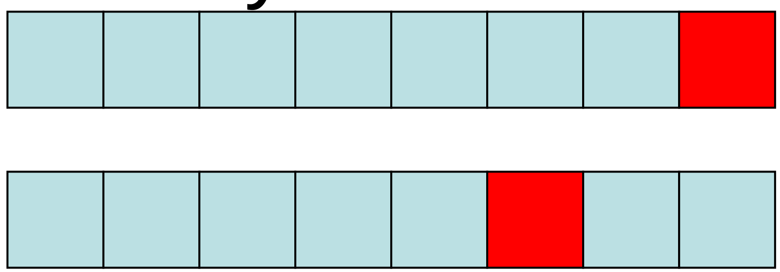
Topological Innovation



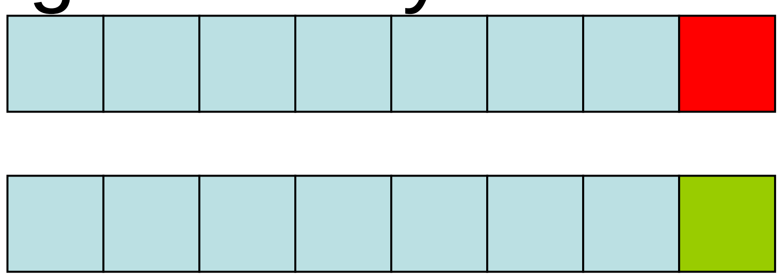
Topology Matching Problem



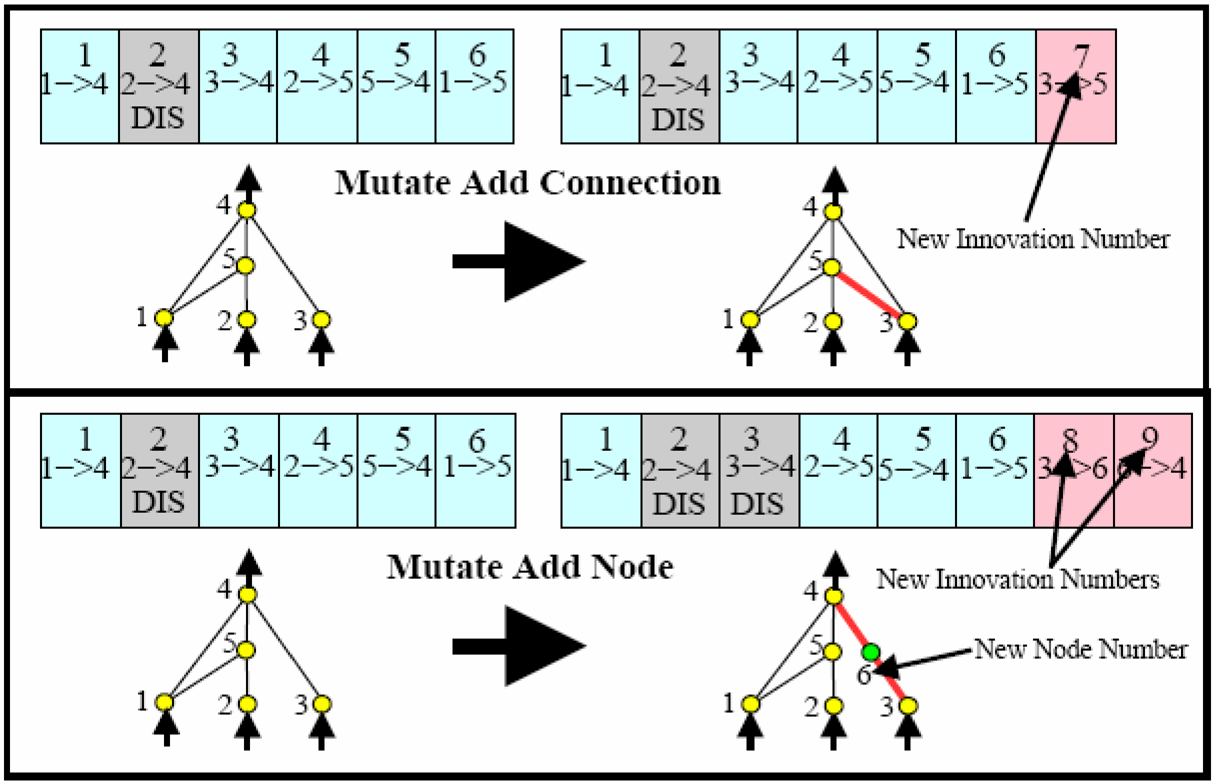
- Problem arises from adding new genes
- Same gene may be in different positions



- Different genes may be in same positions

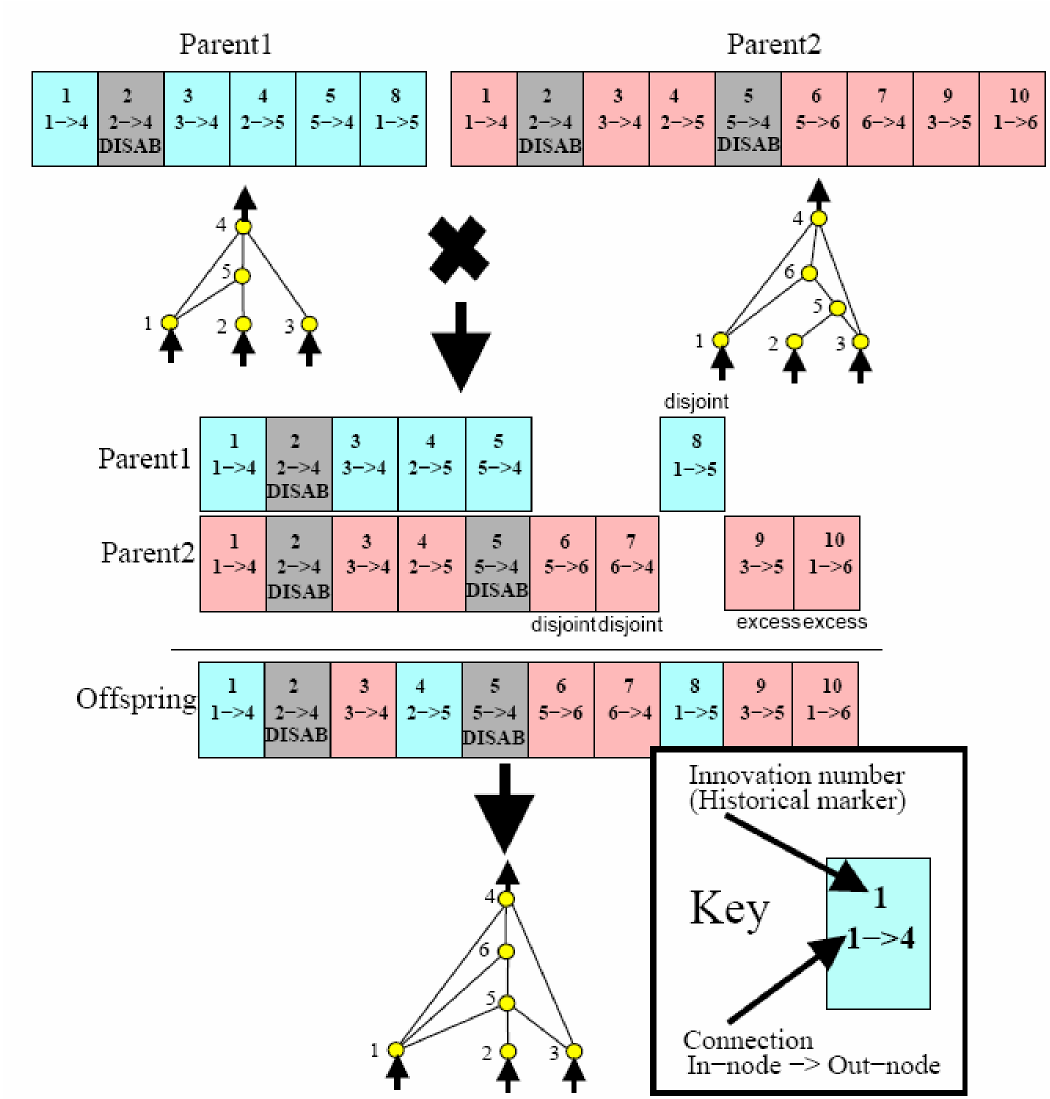


Solution: Tracking Genes through Historical Markings



The numbers tell exactly when in history particular topological features appeared, so now they can be matched up any time in the future. In other words, they reveal *gene homology*.

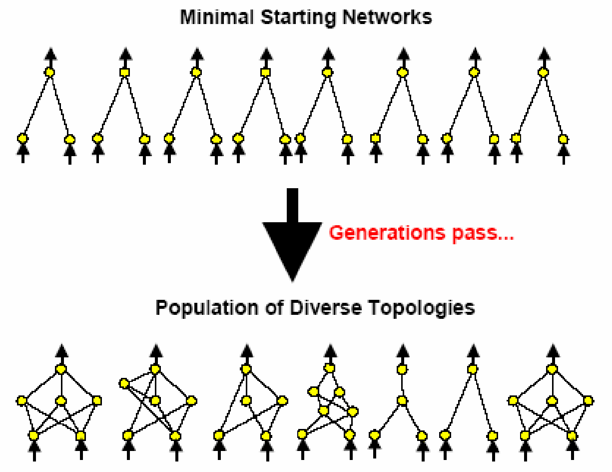
Matching up Genes



Second Component: Speciation Protects Innovation

- Organisms grouped by similarity (compatibility)
- Innovative individuals compete within their niche instead of with the population at large
- Protects innovative solutions so they have time to optimize

Third Component: Complexification from Minimal Structure

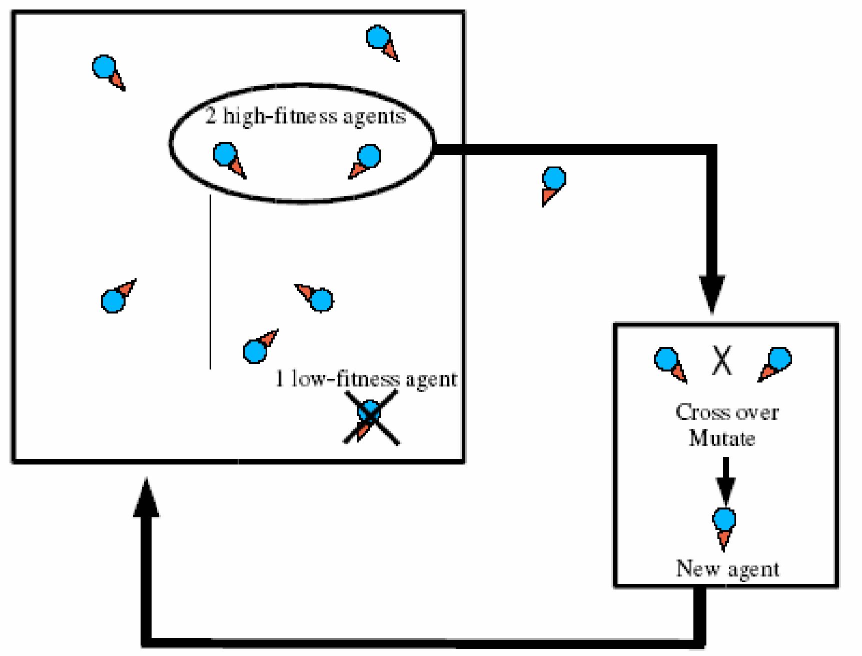


- Search begins in minimal-topology space
- Lower-dimensional structures easily optimized
- Useful innovations eventually survive
- So search transitions into good part of higher-dim. space
- *The ticket to high-dimensional space*

Generations May Not Always Be Appropriate

- When a population is evaluated simultaneously
  - Many are observable at the same time
  - Therefore, entire population would change at once
  - A sudden change is incongruous, highly noticeable
- Players want things to improve constantly

NEAT for Video Games: Real-time NEAT (rtNEAT)



- Real-time replacement loop (non-generational) with NEAT
- Solves several new issues when evolution is real-time
  - Evaluation is asynchronous
  - When to replace?
  - How to assign fitness?
  - Fast enough to change during play


NORMAL TURRET
BETTER TURRET
BRUTAL TURRET
STATIC ENEMY
ROVER
PRETRAINED
1
APPLY
Min
Approach Enemy Range
Max
Near
5
Lifetime Slider (in seconds)
120
18
EVOLUTION
COLLISION
HUD
SPAWN
SAVE ARMY
SAVE MAP
RESET
HELP

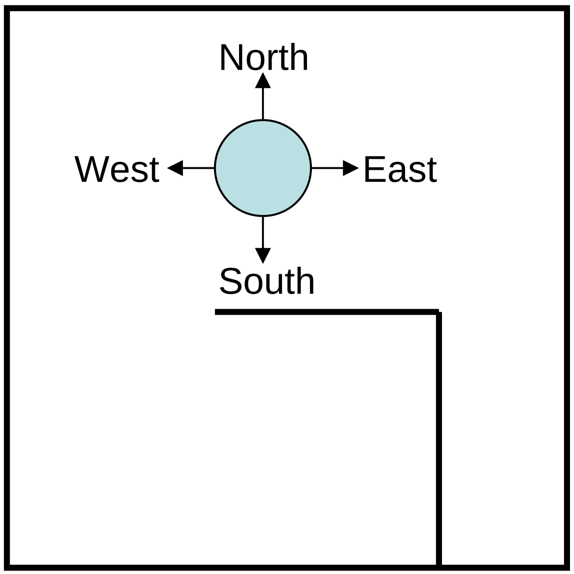
Questions

- How should outputs control the NPC?
- How should sensors detect the world?
- How can it all happen at the same time?

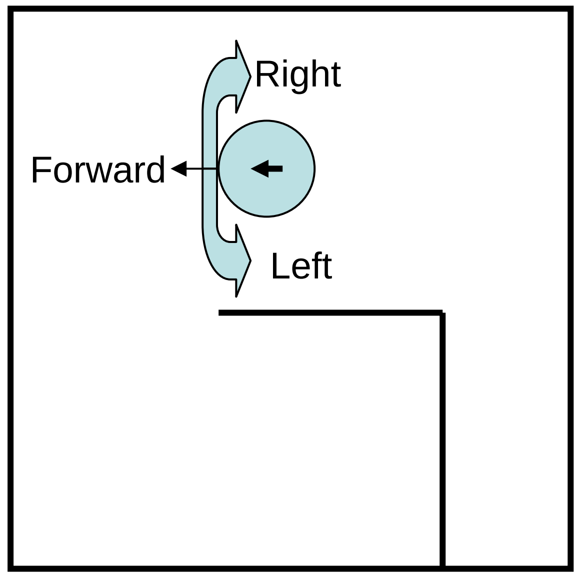
- Neural networks like things a certain way!
  – But it’s an art form to find how that is

Outputs

- First-person / deictic / egocentric
vs. third-person / absolute/ bird's eye view



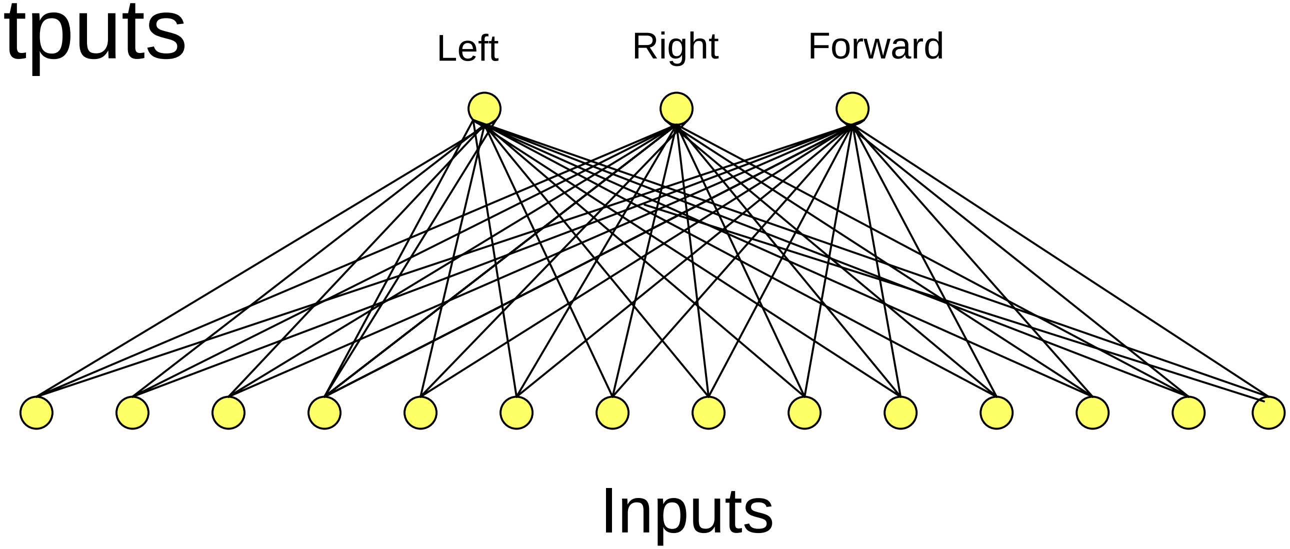
Absolute



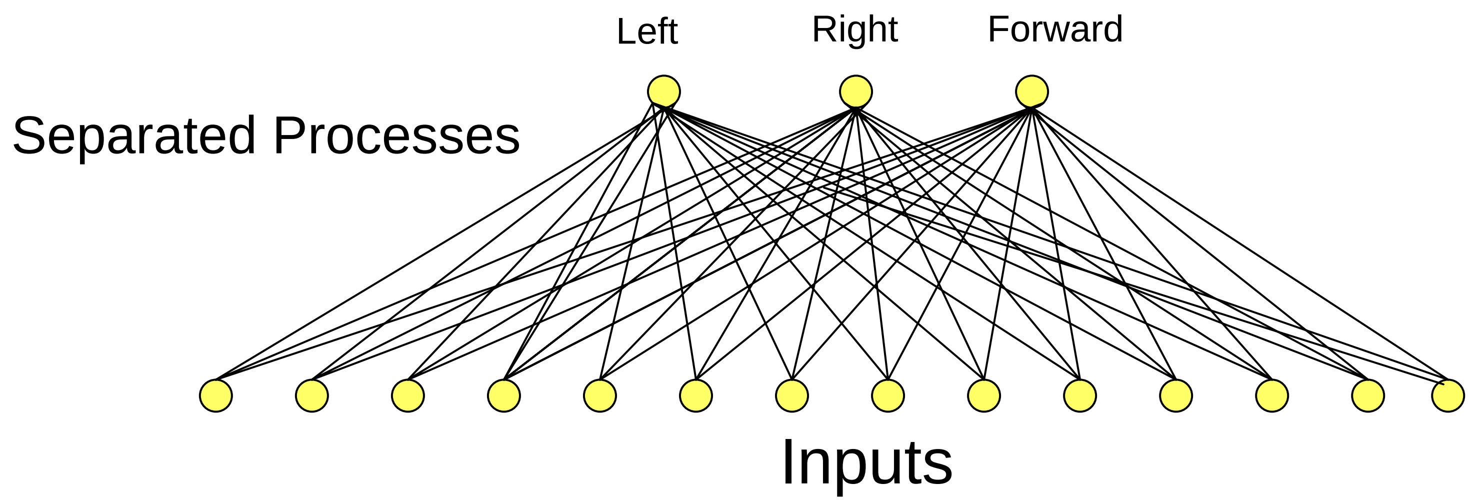
Egocentric

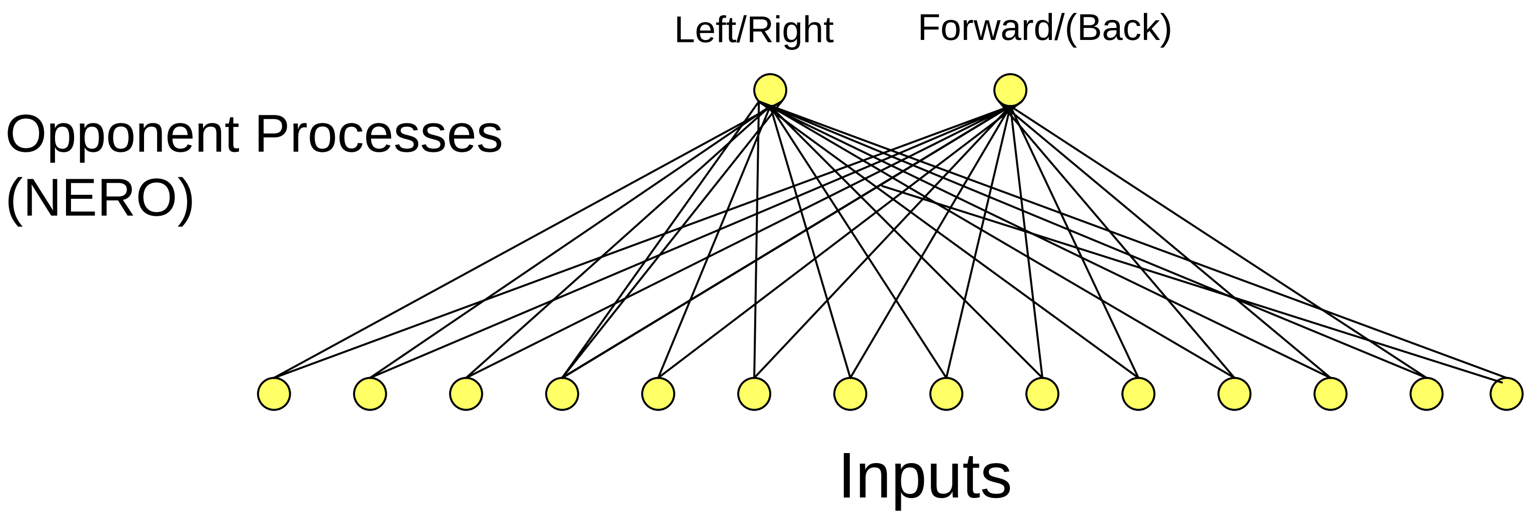
Egocentric Advantages

- Fewer directions to learn
  - Rotational invariance
- Angles of motion need not be discretized
- Fewer outputs
- Intuitive

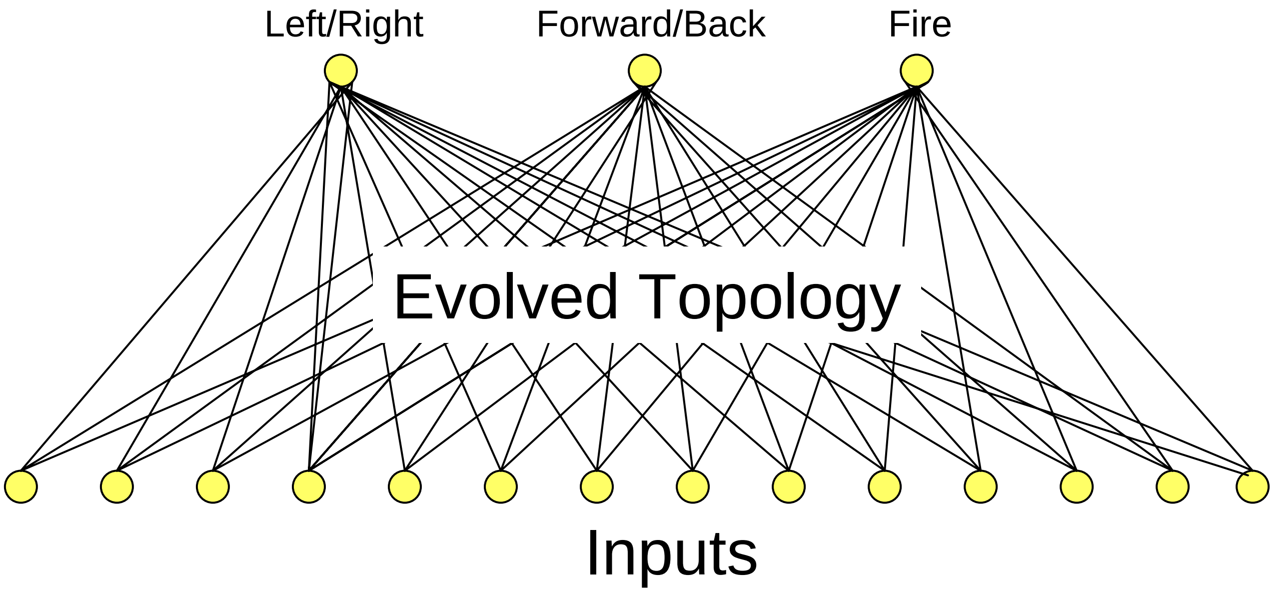


More Decisions!





NERO Outputs



Inputs Are Even Trickier

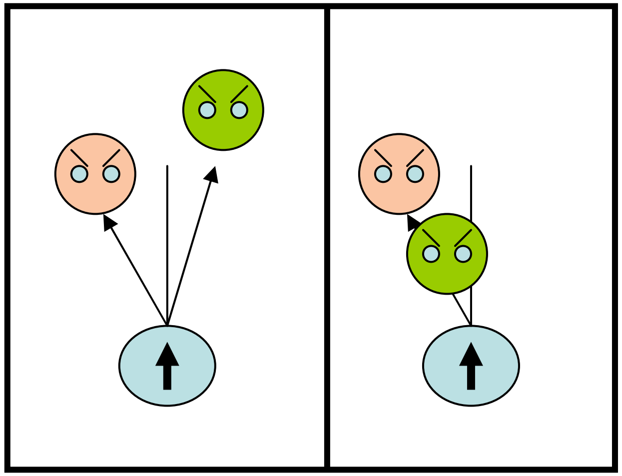
- What kinds of things can I see?
  - Discrete objects (enemies, friends, food)
  - Contiguous objects (walls, obstacles)
  - Implicit objects (line of fire, crowds)
- What coordinate system do I see in?
  - Egocentric/absolute
  - Polar/Cartesian
  - Visual field
- How many inputs can I handle?

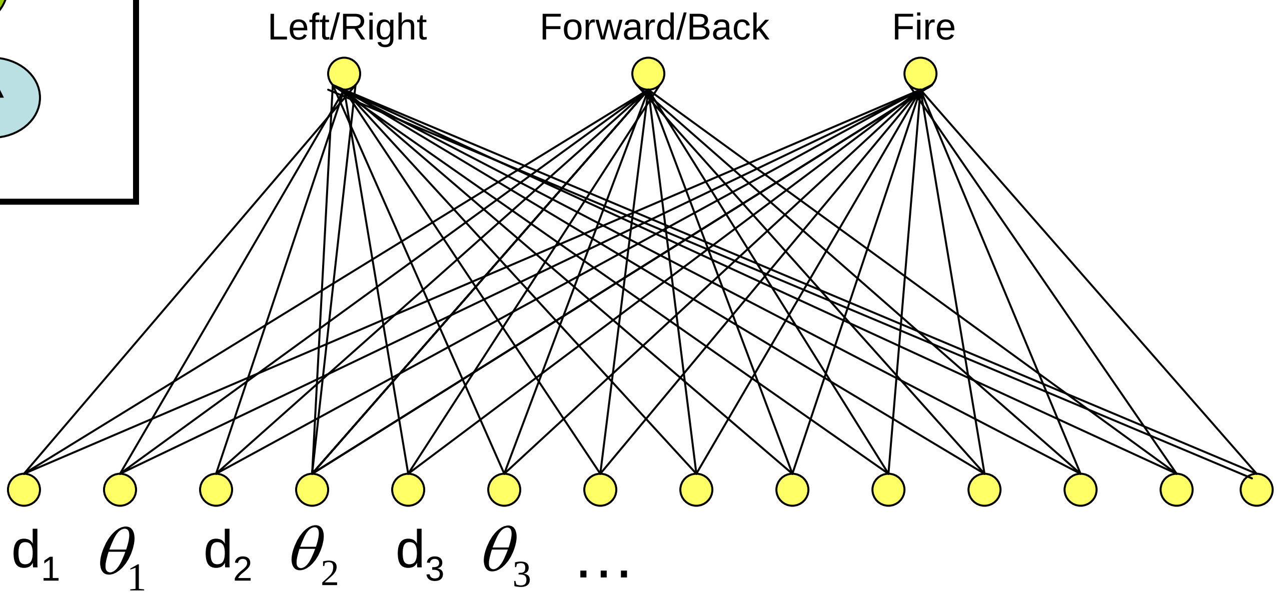
General Rule of Thumb

- Match input and output coordinate system
  - Polar in : Polar out
  - Cartesian in: Cartesian out
  - High should run : Run on high
- In other words…
  - Minimize the need for internal calculations
  - Coordinate transforms require hidden nodes
- So let’s stay egocentric

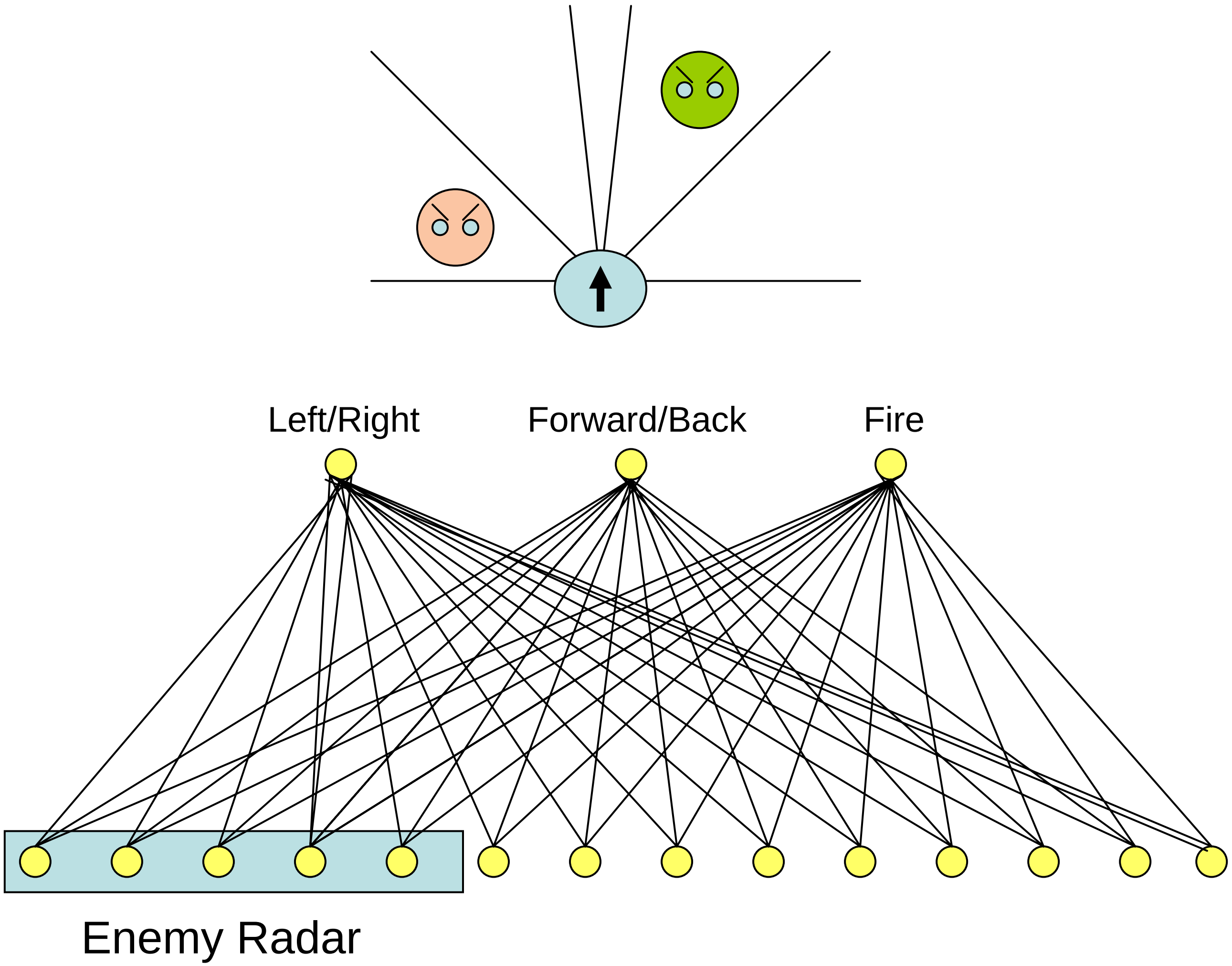
The Problem with Regular Polar

The jumping enemies

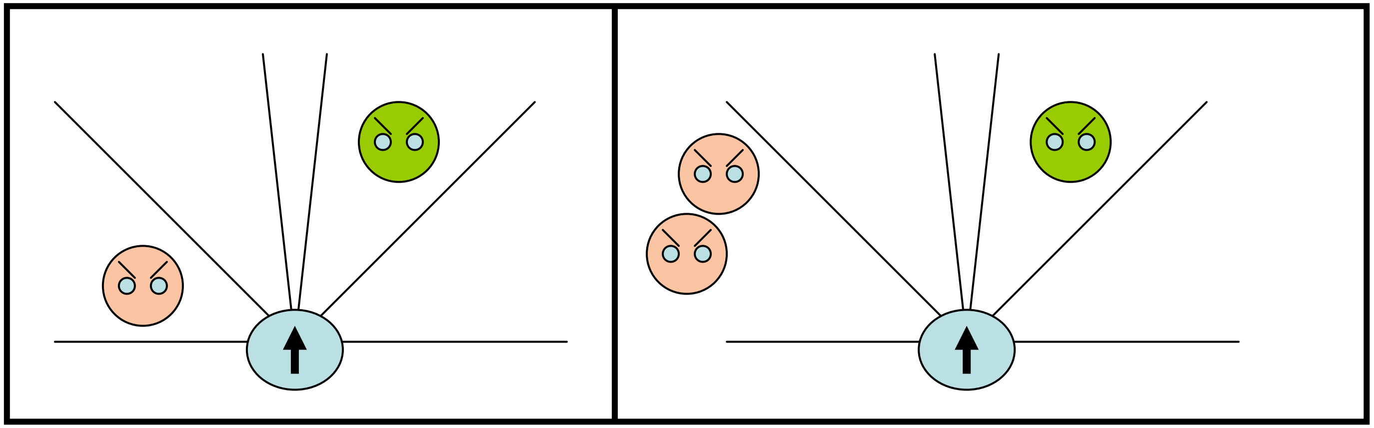


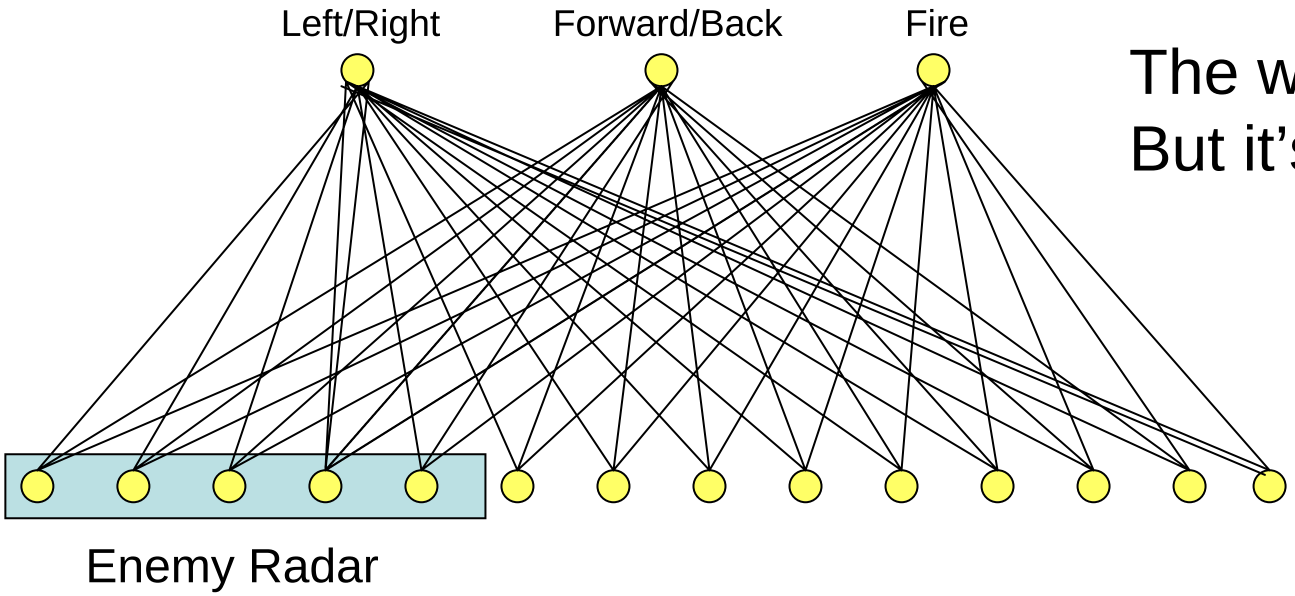


Fixed-Angle Discretized “Radar”



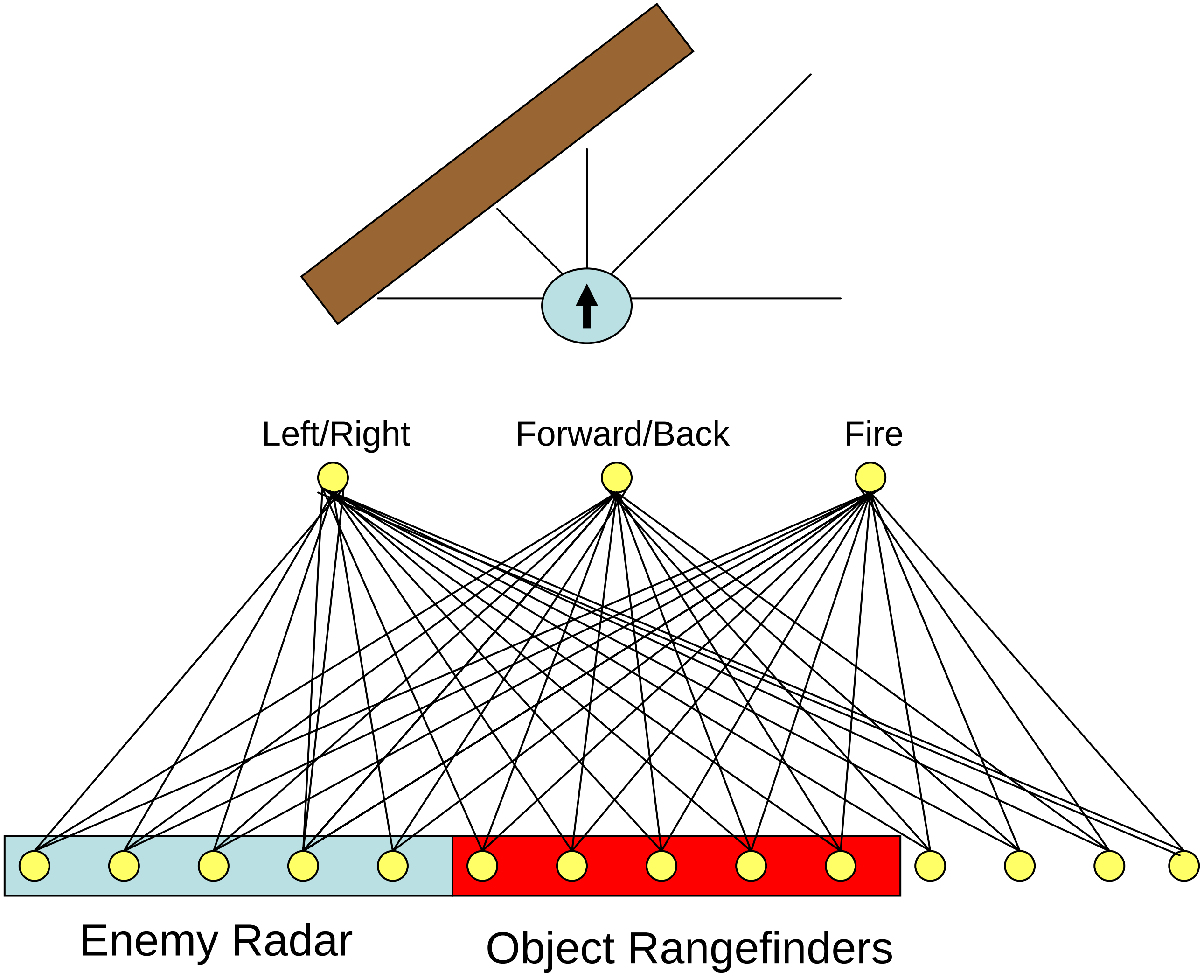
Sensory Aliasing



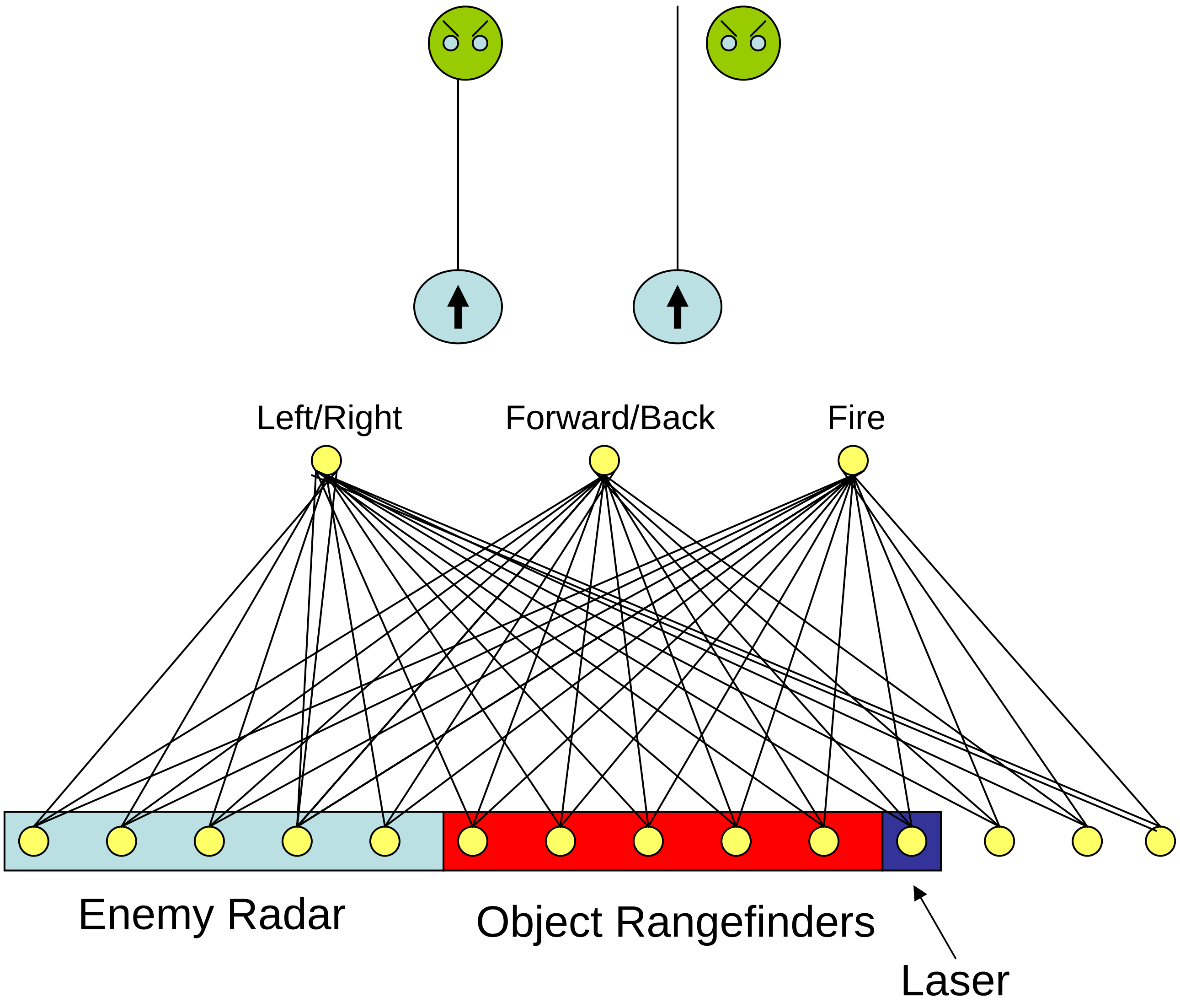


The world is ambiguous
But it’s ok!

Contiguous Objects “Rangefinder”



On Target "Laser Sight"



Do You Really Want to Learn to Shoot?

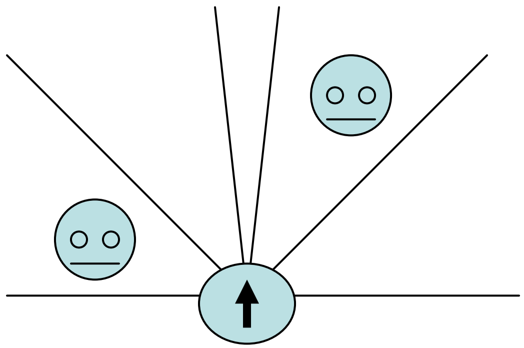
- Learning to shoot when “on target” is not enough
  - Enemies are on the run
  - Depending how fast, accuracy is difficult
- Shooting is more fun with a little “cheat”
  - Prod the gun a couple degrees to the right angle
- Ultra-precision is not an ANN strong suit
  - Not because of non-Markov property

More on Shooting

- People can rotate their arm without rotating their heading
  - Sounds cool but…
  - Introduces more outputs and more degrees of freedom
  - Disconnects the semantics of running from the semantics of shooting: A lot to ask for?
- Traditional to shoot in the direction of heading

I Want to See My Friends!

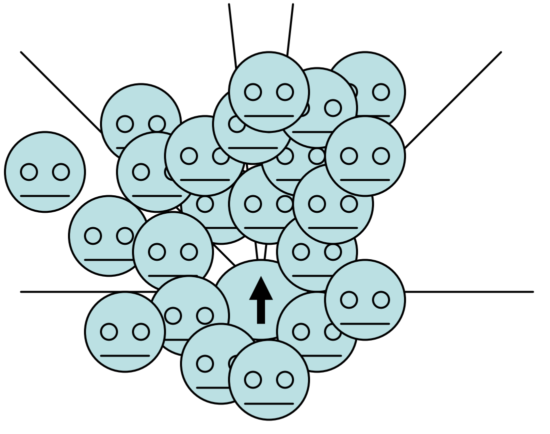
- Can I use “radar?”



- If you have a few friends ok, but…

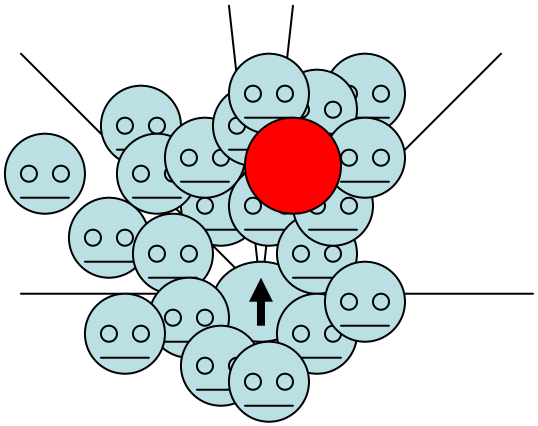
Too Many Friends Are Hard to See

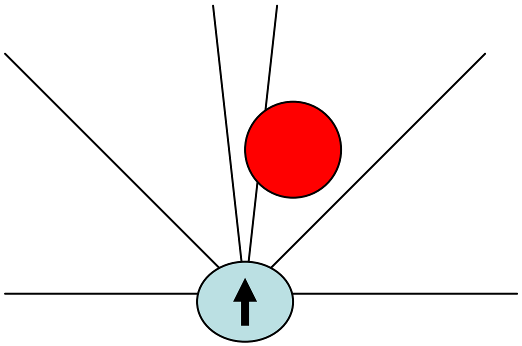
- The downside of popularity



Center of Mass Sensor

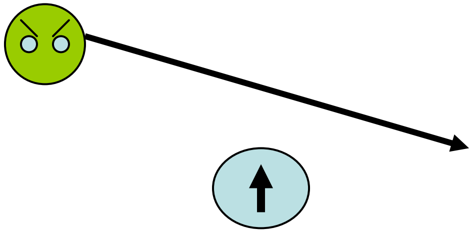
- Now I know what's going on
- Produces "flocking"-like behaviors



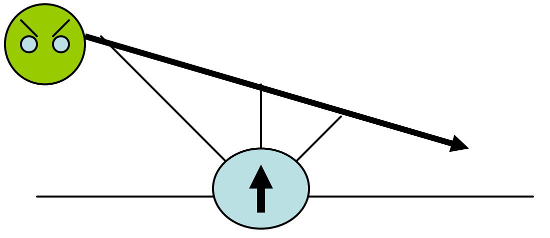


Line-of-Fire Sensors

- Where should I run?



Line-of-Fire is Like a Wall

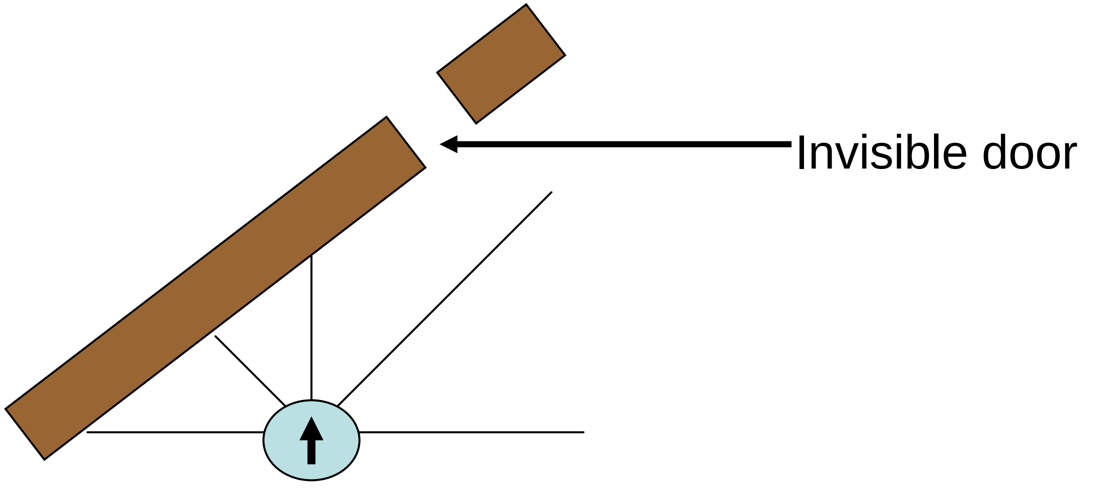


Rangefinders can save lives!

- But don’t *train* with too many enemies
- Need to learn the basic concept from one or two lines at a time

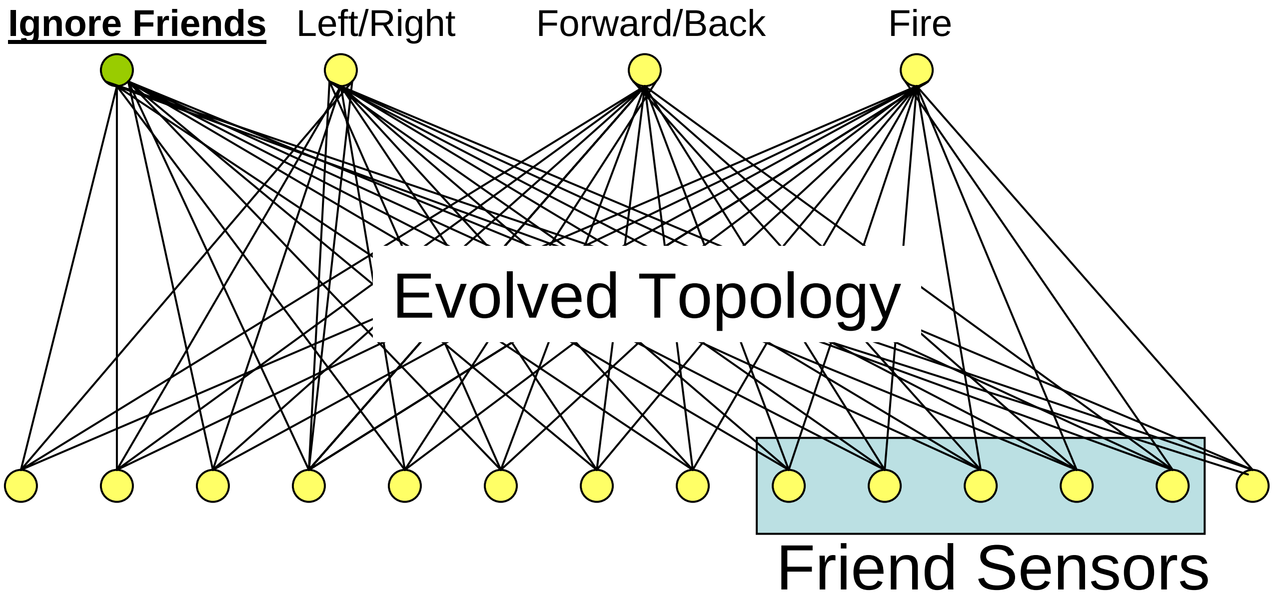
Sensors Are Expensive

- A burden on the CPU
- More weights to learn for the ANN
- The fewer you have the better
- However, resolution needs to be as high as the width of the smallest salient object



Learning to Ignore Things

- A multimodal life is a difficult life
- An unusual kind of output
- When on, zero-out a class of sensors



Multi-Agent Processing

- When 50 agents are on the field…
  - 50 agents see each other and everything else
  - Every sensor checks for everyone at every tick
  - n^2 processing without some tricks
- Occlusion makes it worse
  - Who can't see whom or what?
  - *If you can see through walls, you can run faster too*
- The egocentric world is expensive compared to the bird's eye
  - Some engines better suited than others
    - e.g. sending rays into walls
- Not all sensors need to update at every tick
  - Wall sensors less critical than enemy sensors

Idiosyncratic Behavior

- NEROs like to run in circles
- Such behavior can be stopped
  - Make it physically impossible
  - Penalize it through fitness (not as effective)
  - Make actions more high level

High Level Actions

- Outputs can represent macro-concepts
  - “Run to wall and duck”
  - “Attack the enemy”
  - “Dodge bullets”
- Each output connects to a traditional script
  - Addresses real-world content-control concerns while preserving machine learning
- The danger
  - Loss of creativity
  - The eternal “pleasure” button

Initialization from Scripts

- NEAT or rtNEAT neural networks can be initialized from FSM-like scripts
- Uses ideas from Knowledge-Based ANN (KBANN)
- Requires arbitrary ANN structure
- Evolution can start with traditional scripted behaviors and improve from there

Ryan Cornelius, Kenneth O. Stanley, and Risto Miikkulainen (2006). Constructing Adaptive AI Using Knowledge-Based NeuroEvolution. In Rabin, Woodcock, Forrester, Houlette, Orkin, and Manslow (editors), *AI Game Programming Wisdom 3*. Boston, MA: Charles River Media.

Asynchronous Evaluation: How Many Ticks Between Replacements?

Let M be the minimum ticks alive or age of maturity

Let I be the percentage of the population that is ineligible

Let $|P|$ be the pop. size

Let T be ticks between replacements

- Law of eligibility: $I = \frac{M}{|P|T}$
- So we can give our worst-case tolerable I and derive T:
  $T = \frac{M}{|P|I}$
- $\Rightarrow$ rtNEAT automatically decides T

- Intuitions:
  - The more often replacement occurs, the fewer are eligible
  - The larger the population, the more are eligible
  - The higher the age of maturity, the fewer are eligible

When to Use Neuroevolution

- NPCs can evolve in any game with sufficient interaction
- Or they can even be *pre-evolved* to avoid scripting and predictability
- Enemies or friends
  - Neuroevolution needs only the right fitness criteria

More information

- NERO Page: http://nerogame.org
- My Homepage: http://www.cs.ucf.edu/~kstanley
- NEAT Users Group: http://groups.yahoo.com/group/neat
- Evolutionary Complexity Research Group:
- http://eplex.cs.ucf.edu