

Neural Networks in Supreme Commander 2

Mike “**Sorian**” Robbins
Gameplay Engineer



twitter.com/merobbins

The Four Neural Networks

LAND



NAVAL



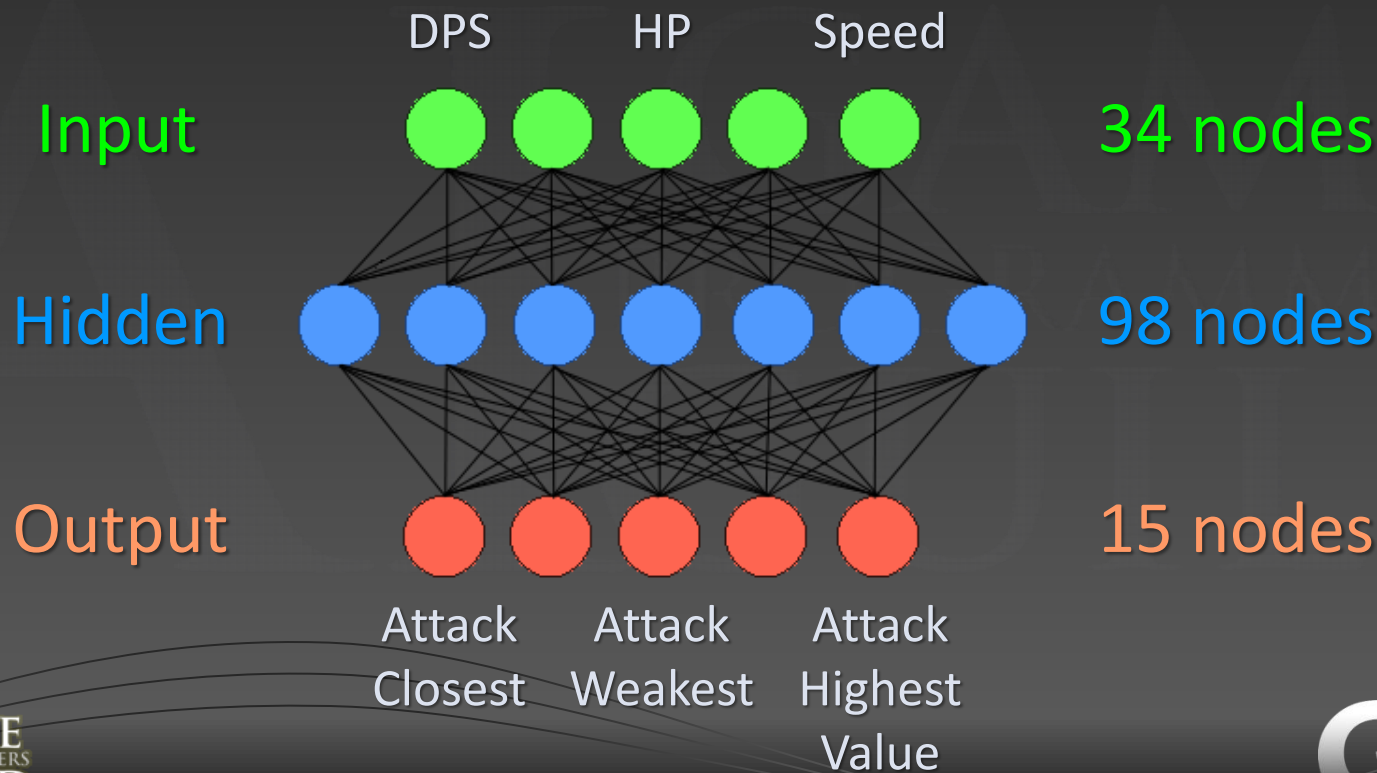
FIGHTER



BOMBER



Back Propagating Neural Network



Fight or Flight

- Platoon encounters enemies.
 - Platoon decides to engage.
- Platoon encounters point defenses.
 - Losses start adding up.
 - Platoon decides to back off.
 - Continues engaging enemy units.
- Enemy reinforcements arrive.
 - Time to cut losses and run.



**So, how does the AI actually use
the neural networks?**

Evaluate Platoon and Enemies

Evaluate 17 data points for each

of Units
HP
Overall DPS
Speed

Mass Value
Shields
of ACUs
Repair Rate

Short Range Static DPS
Mid Range Static DPS
Long Range Static DPS

Short Range Mobile DPS
Mid Range Mobile DPS
Long Range Mobile DPS

Alternate DPS

Two Different Threat Values Based on Platoon Type

Feed Input Into the Neural Network



- Data points are turned into two sets of deltas.
 - Platoon vs Enemies.
 - Enemies vs Platoon.
- Neural network is fed forward.
 - Pushes input through hidden layer and out to output layer.
- Output corresponds to platoon actions.

React Based on Output

Attack Weakest

Attack Closest

Attack Highest Value

Attack Highest Range

Attack Mobile

Attack Engineer

Attack Resource

Attack Shield

Attack Defense

Attack the above
from Range

Attack ACU



If all else fails, run!

The ACU

IF THIS BLOWS UP



GAME OVER!

How do we teach the neural network what the best actions is?

I have *no* idea what the best action is.

Neural Network Benefits

- No need to worry about the apples to oranges problem.
 - How do you compare DPS to speed or health?
- No need to worry about weighting inputs.
 - What input is more important?
- No need to create an algorithm for using the inputs.
- Don't need to worry about inputs not being useful.
 - Is something like speed even useful?

All of this is taken care of during training!

During Training



- Neural network is used mostly the same as in live product.
 - Network is given input.
 - Network is fed forward.
 - However, output is completely ignored.
- Platoons choose actions at random.
- Platoon reports results to neural network.

**How does the platoon tell the network
whether a decision was good or bad?**

Determining Fitness



Did I hurt them more than they
hurt me?

Not Just About HP

- Just because a platoon “wins” a battle does not make it a good decision.
- Just because a platoon “loses” a battle does not make it a bad decision.

Fitness is determined by looking at all of the 17 input values, not just HP.

It's all the same

The fitness function is almost the same process as evaluating enemies to choose an action.

The 17 inputs are compared to the pre-combat values and turned into deltas.

Deltas are compared between enemies and the platoon.

Learning

- Output node adjusted closer to expected output.
- Adjustments are back-propagated.
 - Adjusts weights for child nodes.
- Neural network learns what the output should be.
- Neural network doesn't care what the numbers mean.



Training Time

- 1 hour per network + test time.
- Completely hands off training.
- Any behavior/neural network changes starts training over!
- Neural networks handle balance changes well.

How do you know if they are working?

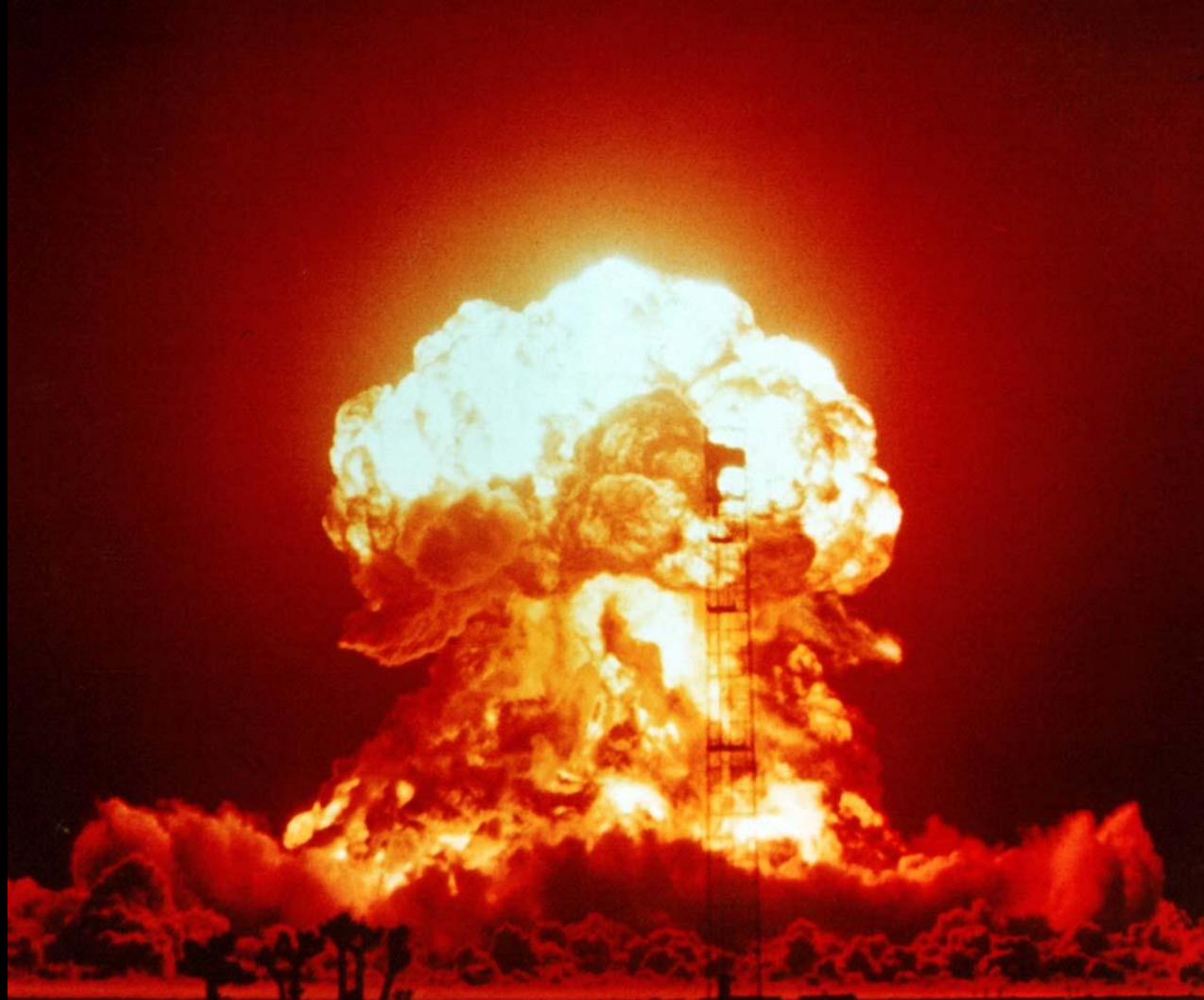
Visual Debugging



Visual Debugging

Visual debugging is crucial!

- Lets you know whether odd behavior is being caused by:
 - Neural network output.
 - Underlying platoon/unit behaviors.
 - **Or something more sinister.**



Sometimes, They are *Too* Smart

Neural networks find the optimal solution.

Losing your entire platoon to take out
one unit is not optimal.

The Infamous ACU



THIS ISSUE HOUNDED ME FOR WEEKS!!!
NO, SERIOUSLY, WEEKS!

Customizing Training

- Was there an ACU there before?
- Did it blow up?
- Were we attacking it?

If all the above is true: Good platoon!



Customizing Training

```
INFO: *AI DEBUG: Army1: Using research index 2
INFO: *AI DEBUG: Using archetype - DefaultRushArchetype
INFO: *AI DEBUG: AI Archetype:Rush Build Bonus: 0 Res Bonus 0 Vet Bonus: 0 Intel Type: None Target Pref: Weakest
INFO: *AI DEBUG: Starting StrategicManager with blueprint - DefaultRushArchetype
INFO: *AI DEBUG: Army2: Using research index 3
INFO: *AI DEBUG: Using Skimish Base Name = DefaultRush
INFO: *AI DEBUG: Using Skimish Base Name = DefaultRush
INFO: *AI DEBUG: Game ended. Dumped Neural Net Data.
INFO: *AI DEBUG: Initializing skimish
INFO: *AI DEBUG: Game started. Training Neural Nets.
INFO: *AI DEBUG: Using archetype - DefaultRushArchetype
INFO: *AI DEBUG: AI Archetype:Rush Build Bonus: 0 Res Bonus 0 Vet Bonus: 0 Intel Type: None Target Pref: Weakest
INFO: *AI DEBUG: Starting StrategicManager with blueprint - DefaultRushArchetype
INFO: *AI DEBUG: Adding Neural Network - LandAttack
INFO: *AI DEBUG: Adding Neural Network - NavalAttack
INFO: *AI DEBUG: Adding Neural Network - BomberAttack
INFO: *AI DEBUG: Adding Neural Network - AirAttack
INFO: *AI DEBUG: Army1: Using research index 3
INFO: *AI DEBUG: Using archetype - DefaultRushArchetype
INFO: *AI DEBUG: AI Archetype:Rush Build Bonus: 0 Res Bonus 0 Vet Bonus: 0 Intel Type: None Target Pref: Weakest
INFO: *AI DEBUG: Starting StrategicManager with blueprint - DefaultRushArchetype
INFO: *AI DEBUG: Army2: Using research index 2
INFO: *AI DEBUG: Using Skimish Base Name = DefaultRush
INFO: *AI DEBUG: Using Skimish Base Name = DefaultRush
INFO: *AI DEBUG: Army 2: Commander overcharging
INFO: *AI DEBUG: Army 2: Commander overcharging
INFO: *AI DEBUG: (Land) I killed an ACU, and I liked it!
INFO: *AI DEBUG: Game ended. Dumped Neural Net Data.
```

**"I KILLED AN ACU,
AND I LIKED IT!"**

ACU Assassins



Takeaway

Neural networks are like any other AI tool

If you use them for something they are good at
the results can be amazing and dynamic!



twitter.com/merobbins