

neuroevolution-class

March 2, 2025

1 Neuroevolution class

In this class we will investigate the use of Neat to solve the [Gym Lunar Lander game](#).

First execute the following code which is used to render Gym environments in jupyter:

```
[1]: # Virtual display
from pyvirtualdisplay import Display

virtual_display = Display(visible=0, size=(14, 9))
virtual_display.start()
import matplotlib.pyplot as plt
%matplotlib inline
from IPython import display

def jrender(env, step=0, info=""):
    plt.figure("display", (100, 3))
    plt.clf()
    plt.imshow(env.render())
    plt.title("%s | Step: %d | %s" % (env.spec.id, step, info))
    plt.axis('off')

    display.clear_output(wait=True)
    display.display(plt.gcf())
    plt.close()
```

1.1 Task 1: Gym

Create an instance of the `LunarLander-v2` modifying the following example (modified from https://www.gymnasium.dev/content/basic_usage/)

Before running the code make the following changes: - change `render_mode` to “`rgb_array`” - modify the code to run for 200 steps - add `jrender(env)` as the last instruction of the for loop to update the screen - pass the step number as second parameter to the `jrender`

```
[2]: import gym
env = gym.make("LunarLander-v2", render_mode="human")

observation, info = env.reset()
```

```

for _ in range(1000):
    observation, reward, terminated, truncated, info = env.step(env.
    ↪action_space.sample())

    if terminated or truncated:
        observation, info = env.reset()

env.close()

```

```

/opt/jupyterhub/pyvenv/lib/python3.10/site-
packages/gym/utils/passive_env_checker.py:233: DeprecationWarning: `np.bool8` is
a deprecated alias for `np.bool_`. (Deprecated NumPy 1.24)
    if not isinstance(terminated, (bool, np.bool8)):

```

```

[ ]: import gym

# Create the LunarLander environment with the required render mode
env = gym.make("LunarLander-v2", render_mode="rgb_array")

observation, info = env.reset()

for step in range(200): # Run for 200 steps
    action = env.action_space.sample() # Random action
    observation, reward, terminated, truncated, info = env.step(action)

    jrender(env, step) # Update the screen with step number

    if terminated or truncated:
        break # Stop when the episode ends

env.close()

```

[]: Ellipsis

Make the following changes to the code: - remove the if statement (including the body) and change the for into a while loop to run the environment while the step method returns not terminated and not truncated - pass as third parameter to jrender the total reward accumulated so far. You should see the reward decreases of about 100 points when crashing. - to speedup the visualisation you can change the loop to call jrender every 5 frames or when terminated or truncated (to show the final reward)

```

[ ]: import gym

# Create the LunarLander environment with the required render mode
env = gym.make("LunarLander-v2", render_mode="rgb_array")

```

```

observation, info = env.reset()
total_reward = 0 # Track total accumulated reward
step = 0

while True: # Run until the environment terminates
    action = env.action_space.sample() # Random action
    observation, reward, terminated, truncated, info = env.step(action)
    total_reward += reward # Update total reward
    step += 1

    if step % 5 == 0 or terminated or truncated: # Update screen every 5
        ↪frames or at the end
        jrender(env, step, f"Total Reward: {total_reward:.2f}")

    if terminated or truncated:
        break # Stop when the episode ends

env.close()

```

[]: Ellipsis

Refresh the different types of observation and action spaces at https://www.gymnasium.dev/content/basic_usage/#spaces . Print out the environment observation and action space.

```

[ ]: import gym

# Create the LunarLander environment
env = gym.make("LunarLander-v2", render_mode="rgb_array")

# Print observation and action space types
print("Observation Space:", env.observation_space)
print("Action Space:", env.action_space)

observation, info = env.reset()
total_reward = 0 # Track total accumulated reward
step = 0

while True: # Run until the environment terminates
    action = env.action_space.sample() # Random action
    observation, reward, terminated, truncated, info = env.step(action)
    total_reward += reward # Update total reward
    step += 1

    if step % 5 == 0 or terminated or truncated: # Update screen every 5
        ↪frames or at the end
        jrender(env, step, f"Total Reward: {total_reward:.2f}")

```

```

    if terminated or truncated:
        break # Stop when the episode ends

env.close()

```

[]: Ellipsis

What does each value in the observation and action space represent? Consult the documentation of [Lunar Lander](#) to find out.

2 LunarLander Observation and Action Space Explanation

2.1 Observation Space

The observation space consists of an array with **8 continuous values**, representing different aspects of the lander's state:

Index	Description
0	X position (horizontal coordinate)
1	Y position (vertical coordinate)
2	X velocity (horizontal speed)
3	Y velocity (vertical speed)
4	Lander's angle (rotation)
5	Angular velocity (rotation speed)
6	Left landing leg contact with ground (1 if touching, else 0)
7	Right landing leg contact with ground (1 if touching, else 0)

2.2 Action Space

The action space consists of **4 discrete actions**:

Action Index	Description
0	Do nothing
1	Fire left engine
2	Fire main engine (up)
3	Fire right engine

The goal is to **control the lander** to land smoothly on the designated landing pad using these actions while minimizing fuel usage.

2.3 Task 2: Neat

We are now experimenting with the basic XOR Neat example.

- Download the configuration file <https://raw.githubusercontent.com/CodeReclaimers/neat-python/master/examples/xor/config-feedforward-partial> by running the following

```
[7]: ! wget https://raw.githubusercontent.com/CodeReclaimers/neat-python/master/
      ↪ examples/xor/config-feedforward-partial
```

```
--2025-02-10 12:47:57-- https://raw.githubusercontent.com/CodeReclaimers/neat-
python/master/examples/xor/config-feedforward-partial
Resolving raw.githubusercontent.com (raw.githubusercontent.com)...
2606:50c0:8002::154, 2606:50c0:8003::154, 2606:50c0:8000::154, ...
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|2606:50c0:8002::154|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1931 (1.9K) [text/plain]
Saving to: 'config-feedforward-partial'

config-feedforward- 100%[=====>] 1.89K --.-KB/s in 0s

2025-02-10 12:47:57 (13.8 MB/s) - 'config-feedforward-partial' saved [1931/1931]
```

You can edit the downloaded file by clicking on it from the jupyter home.

Execute the following code (modified version) from the base XOR Neat example at https://neat-python.readthedocs.io/en/latest/xor_example.html

```
[8]: """
    2-input XOR example -- this is most likely the simplest possible example.
    """

    from __future__ import print_function
    import os
    import neat
    # import visualize

    # 2-input XOR inputs and expected outputs.
    xor_inputs = [(0.0, 0.0), (0.0, 1.0), (1.0, 0.0), (1.0, 1.0)]
    xor_outputs = [ (0.0,), (1.0,), (1.0,), (0.0,) ]

    def eval_genomes(genomes, config):
        for genome_id, genome in genomes:
            genome.fitness = 4.0
            net = neat.nn.FeedForwardNetwork.create(genome, config)
            for xi, xo in zip(xor_inputs, xor_outputs):
                output = net.activate(xi)
                genome.fitness -= (output[0] - xo[0]) ** 2

    def run(config_file):
        # Load configuration.
```

```

config = neat.Config(neat.DefaultGenome, neat.DefaultReproduction,
                     neat.DefaultSpeciesSet, neat.DefaultStagnation,
                     config_file)

# Create the population, which is the top-level object for a NEAT run.
p = neat.Population(config)

# Add a stdout reporter to show progress in the terminal.
p.add_reporter(neat.StdOutReporter(True))
# stats = neat.StatisticsReporter()
# p.add_reporter(stats)
# p.add_reporter(neat.Checkpointer(5))

# Run for up to 300 generations.
winner = p.run(eval_genomes, 300)

# Display the winning genome.
print('\nBest genome:\n{!s}'.format(winner))

# Show output of the most fit genome against training data.
print('\nOutput:')
winner_net = neat.nn.FeedForwardNetwork.create(winner, config)
for xi, xo in zip(xor_inputs, xor_outputs):
    output = winner_net.activate(xi)
    print("input {!r}, expected output {!r}, got {!r}".format(xi, xo,
↪output))

```

Execute the run function passing the configuration file `config-feedforward-partial` downloaded before.

```
[9]: run("config-feedforward-partial")
```

```

***** Running generation 0 *****

Population's average fitness: 2.28860 stdev: 0.26536
Best fitness: 2.98951 - size: (1, 1) - species 1 - id 73
Average adjusted fitness: 0.288
Mean genetic distance 1.749, standard deviation 0.557
Population of 150 members in 2 species:
  ID   age  size  fitness  adj fit  stag
  ====  ===  ====  =====  =====  ====
    1    0   77     3.0    0.336    0
    2    0   73     2.9    0.240    0
Total extinctions: 0
Generation time: 0.021 sec

***** Running generation 1 *****

```

Population's average fitness: 2.21862 stdev: 0.30503
 Best fitness: 2.98951 - size: (1, 1) - species 1 - id 73
 Average adjusted fitness: 0.217
 Mean genetic distance 1.757, standard deviation 0.638
 Population of 150 members in 2 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	1	77	3.0	0.271	1
2	1	73	2.9	0.163	1

Total extinctions: 0
 Generation time: 0.019 sec (0.020 average)

***** Running generation 2 *****

Population's average fitness: 2.25302 stdev: 0.31889
 Best fitness: 2.99930 - size: (2, 4) - species 1 - id 326
 Average adjusted fitness: 0.362
 Mean genetic distance 1.704, standard deviation 0.642
 Population of 150 members in 2 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	2	60	3.0	0.365	0
2	2	90	3.0	0.359	0

Total extinctions: 0
 Generation time: 0.021 sec (0.021 average)

***** Running generation 3 *****

Population's average fitness: 2.24412 stdev: 0.30935
 Best fitness: 2.99930 - size: (2, 4) - species 1 - id 326
 Average adjusted fitness: 0.386
 Mean genetic distance 1.625, standard deviation 0.522
 Population of 150 members in 2 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	3	59	3.0	0.404	1
2	3	91	3.0	0.369	0

Total extinctions: 0
 Generation time: 0.022 sec (0.021 average)

***** Running generation 4 *****

Population's average fitness: 2.26983 stdev: 0.36125
 Best fitness: 2.99930 - size: (2, 4) - species 1 - id 326
 Average adjusted fitness: 0.635
 Mean genetic distance 1.707, standard deviation 0.512
 Population of 150 members in 2 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	4	53	3.0	0.655	2
2	4	97	3.0	0.615	0

Total extinctions: 0

Generation time: 0.023 sec (0.021 average)

***** Running generation 5 *****

Population's average fitness: 2.27822 stdev: 0.34368

Best fitness: 2.99999 - size: (2, 4) - species 1 - id 863

Average adjusted fitness: 0.285

Mean genetic distance 1.748, standard deviation 0.444

Population of 150 members in 2 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	5	51	3.0	0.310	0
2	5	99	3.0	0.261	0

Total extinctions: 0

Generation time: 0.025 sec (0.022 average)

***** Running generation 6 *****

Population's average fitness: 2.30217 stdev: 0.38208

Best fitness: 2.99999 - size: (2, 4) - species 1 - id 863

Average adjusted fitness: 0.562

Mean genetic distance 1.765, standard deviation 0.448

Population of 150 members in 2 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	6	50	3.0	0.604	1
2	6	100	3.0	0.519	1

Total extinctions: 0

Generation time: 0.025 sec (0.022 average)

***** Running generation 7 *****

Population's average fitness: 2.35169 stdev: 0.40705

Best fitness: 3.00000 - size: (2, 4) - species 1 - id 1125

Average adjusted fitness: 0.603

Mean genetic distance 1.788, standard deviation 0.405

Population of 150 members in 2 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	7	46	3.0	0.598	0
2	7	104	3.0	0.608	0

Total extinctions: 0

Generation time: 0.023 sec (0.022 average)

***** Running generation 8 *****

Population's average fitness: 2.45003 stdev: 0.43999
Best fitness: 3.00000 - size: (2, 4) - species 1 - id 1125
Average adjusted fitness: 0.712
Mean genetic distance 1.755, standard deviation 0.444
Population of 150 members in 2 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	8	52	3.0	0.699	1
2	8	98	3.0	0.724	0

Total extinctions: 0
Generation time: 0.024 sec (0.023 average)

***** Running generation 9 *****

Population's average fitness: 2.42337 stdev: 0.44305
Best fitness: 3.00000 - size: (2, 4) - species 1 - id 1460
Average adjusted fitness: 0.690
Mean genetic distance 1.808, standard deviation 0.446
Population of 150 members in 2 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	9	58	3.0	0.698	0
2	9	92	3.0	0.681	0

Total extinctions: 0
Generation time: 0.024 sec (0.023 average)

***** Running generation 10 *****

Population's average fitness: 2.39683 stdev: 0.44978
Best fitness: 3.00000 - size: (2, 4) - species 1 - id 1460
Average adjusted fitness: 0.643
Mean genetic distance 1.989, standard deviation 0.704
Population of 150 members in 3 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	10	72	3.0	0.637	1
2	10	72	3.0	0.649	0
3	0	6	--	--	0

Total extinctions: 0
Generation time: 0.024 sec (0.023 average)

***** Running generation 11 *****

Population's average fitness: 2.41918 stdev: 0.44942
Best fitness: 3.00000 - size: (2, 4) - species 1 - id 1460

Average adjusted fitness: 0.636

Mean genetic distance 2.108, standard deviation 0.731

Population of 150 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	11	67	3.0	0.677	2
2	11	48	3.0	0.686	0
3	1	25	3.0	0.544	0
4	0	10	--	--	0

Total extinctions: 0

Generation time: 0.026 sec (0.024 average)

***** Running generation 12 *****

Population's average fitness: 2.37349 stdev: 0.44239

Best fitness: 3.00000 - size: (3, 4) - species 4 - id 1887

Average adjusted fitness: 0.364

Mean genetic distance 2.173, standard deviation 0.758

Population of 150 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	12	41	3.0	0.482	3
2	12	51	3.0	0.326	1
3	2	25	3.0	0.108	1
4	1	33	3.0	0.541	0

Total extinctions: 0

Generation time: 0.028 sec (0.024 average)

***** Running generation 13 *****

Population's average fitness: 2.43894 stdev: 0.49691

Best fitness: 3.00000 - size: (2, 2) - species 2 - id 2040

Average adjusted fitness: 0.712

Mean genetic distance 2.267, standard deviation 0.744

Population of 150 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	13	25	3.0	0.735	4
2	13	23	3.0	0.711	0
3	3	27	3.0	0.600	0
4	2	75	3.0	0.804	0

Total extinctions: 0

Generation time: 0.124 sec (0.035 average)

***** Running generation 14 *****

Population's average fitness: 2.48412 stdev: 0.49796

Best fitness: 3.00000 - size: (3, 3) - species 4 - id 2135

Average adjusted fitness: 0.694

Mean genetic distance 2.244, standard deviation 0.752

Population of 151 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	14	52	3.0	0.683	5
2	14	27	3.0	0.753	1
3	4	23	3.0	0.525	1
4	3	49	3.0	0.815	0

Total extinctions: 0

Generation time: 0.029 sec (0.035 average)

***** Running generation 15 *****

Population's average fitness: 2.45355 stdev: 0.49217

Best fitness: 3.00000 - size: (2, 2) - species 4 - id 2325

Average adjusted fitness: 0.697

Mean genetic distance 2.324, standard deviation 0.952

Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	15	47	3.0	0.757	0
2	15	35	3.0	0.556	2
3	5	20	3.0	0.652	2
4	4	14	3.0	0.823	0
5	0	34	--	--	0

Total extinctions: 0

Generation time: 0.031 sec (0.036 average)

***** Running generation 16 *****

Population's average fitness: 2.41412 stdev: 0.46661

Best fitness: 3.00000 - size: (2, 2) - species 5 - id 2445

Average adjusted fitness: 0.593

Mean genetic distance 2.258, standard deviation 0.903

Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	16	62	3.0	0.597	1
2	16	17	3.0	0.600	3
3	6	23	3.0	0.673	3
4	5	20	3.0	0.445	1
5	1	28	3.0	0.649	0

Total extinctions: 0

Generation time: 0.031 sec (0.036 average)

***** Running generation 17 *****

Population's average fitness: 2.44837 stdev: 0.49120
 Best fitness: 3.00000 - size: (2, 3) - species 5 - id 2583
 Average adjusted fitness: 0.679
 Mean genetic distance 2.317, standard deviation 0.935
 Population of 151 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	17	43	3.0	0.711	2
2	17	24	3.0	0.590	4
3	7	24	3.0	0.704	0
4	6	34	3.0	0.618	2
5	2	26	3.0	0.772	0

Total extinctions: 0
 Generation time: 0.033 sec (0.037 average)

***** Running generation 18 *****

Population's average fitness: 2.46673 stdev: 0.48613
 Best fitness: 3.00000 - size: (2, 3) - species 5 - id 2583
 Average adjusted fitness: 0.725
 Mean genetic distance 2.275, standard deviation 0.857
 Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	18	35	3.0	0.767	0
2	18	32	3.0	0.575	5
3	8	25	3.0	0.757	1
4	7	9	3.0	0.710	3
5	3	49	3.0	0.814	1

Total extinctions: 0
 Generation time: 0.032 sec (0.038 average)

***** Running generation 19 *****

Population's average fitness: 2.46709 stdev: 0.50951
 Best fitness: 3.00000 - size: (2, 3) - species 5 - id 2583
 Average adjusted fitness: 0.718
 Mean genetic distance 2.293, standard deviation 0.951
 Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	19	31	3.0	0.768	1
2	19	25	3.0	0.626	6
3	9	26	3.0	0.706	0
4	8	27	3.0	0.688	4
5	4	41	3.0	0.801	2

Total extinctions: 0
 Generation time: 0.032 sec (0.039 average)

***** Running generation 20 *****

Population's average fitness: 2.44654 stdev: 0.49241
Best fitness: 3.00000 - size: (4, 4) - species 5 - id 3000
Average adjusted fitness: 0.716
Mean genetic distance 2.351, standard deviation 1.026
Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	20	17	3.0	0.824	2
2	20	17	3.0	0.717	7
3	10	22	3.0	0.646	0
4	9	60	3.0	0.637	5
5	5	34	3.0	0.757	0

Total extinctions: 0
Generation time: 0.031 sec (0.040 average)

***** Running generation 21 *****

Population's average fitness: 2.39941 stdev: 0.47186
Best fitness: 3.00000 - size: (5, 4) - species 5 - id 3144
Average adjusted fitness: 0.629
Mean genetic distance 2.291, standard deviation 0.957
Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	21	19	3.0	0.690	0
2	21	51	3.0	0.694	8
3	11	21	3.0	0.540	1
4	10	32	3.0	0.563	6
5	6	25	3.0	0.659	0
6	0	2	--	--	0

Total extinctions: 0
Generation time: 0.032 sec (0.040 average)

***** Running generation 22 *****

Population's average fitness: 2.39441 stdev: 0.48706
Best fitness: 3.00000 - size: (5, 5) - species 5 - id 3287
Average adjusted fitness: 0.634
Mean genetic distance 2.414, standard deviation 0.810
Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	22	13	3.0	0.538	1
2	22	54	3.0	0.782	9
3	12	20	3.0	0.612	2

4	11	52	3.0	0.580	7
5	7	2	3.0	0.813	0
6	1	9	2.0	0.481	0

Total extinctions: 0

Generation time: 0.038 sec (0.041 average)

***** Running generation 23 *****

Population's average fitness: 2.39608 stdev: 0.48868

Best fitness: 3.00000 - size: (5, 5) - species 2 - id 3287

Average adjusted fitness: 0.558

Mean genetic distance 2.530, standard deviation 0.808

Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	23	24	3.0	0.558	2
2	23	5	3.0	0.543	0
3	13	15	3.0	0.677	0
4	12	48	3.0	0.701	8
5	8	26	2.1	0.385	1
6	2	32	3.0	0.483	0

Total extinctions: 0

Generation time: 0.039 sec (0.033 average)

***** Running generation 24 *****

Population's average fitness: 2.43969 stdev: 0.48533

Best fitness: 3.00000 - size: (5, 5) - species 6 - id 3287

Average adjusted fitness: 0.690

Mean genetic distance 2.589, standard deviation 0.772

Population of 151 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	24	16	3.0	0.703	3
2	24	25	2.0	0.499	1
3	14	21	3.0	0.670	1
4	13	45	3.0	0.674	9
5	9	31	3.0	0.813	2
6	3	13	3.0	0.780	0

Total extinctions: 0

Generation time: 0.039 sec (0.034 average)

***** Running generation 25 *****

Population's average fitness: 2.55981 stdev: 0.49487

Best fitness: 3.00000 - size: (5, 4) - species 2 - id 3565

Average adjusted fitness: 0.766

Mean genetic distance 2.652, standard deviation 0.885

Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	25	33	3.0	0.837	4
2	25	3	3.0	0.577	2
3	15	21	3.0	0.731	0
4	14	25	3.0	0.834	10
5	10	37	3.0	0.834	3
6	4	31	3.0	0.782	1

Total extinctions: 0

Generation time: 0.040 sec (0.035 average)

***** Running generation 26 *****

Population's average fitness: 2.50957 stdev: 0.51130

Best fitness: 3.00000 - size: (5, 5) - species 6 - id 3695

Average adjusted fitness: 0.690

Mean genetic distance 2.747, standard deviation 0.964

Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	26	37	3.0	0.704	0
2	26	15	2.0	0.327	3
3	16	26	3.0	0.733	1
4	15	14	3.0	0.898	11
5	11	14	3.0	0.743	4
6	5	44	3.0	0.737	2

Total extinctions: 0

Generation time: 0.039 sec (0.035 average)

***** Running generation 27 *****

Population's average fitness: 2.50381 stdev: 0.50333

Best fitness: 3.00000 - size: (5, 5) - species 6 - id 3695

Average adjusted fitness: 0.704

Mean genetic distance 2.779, standard deviation 1.023

Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	27	33	3.0	0.826	0
2	27	17	3.0	0.603	4
3	17	26	3.0	0.648	2
4	16	19	3.0	0.675	12
5	12	7	3.0	0.603	5
6	6	48	3.0	0.865	3

Total extinctions: 0

Generation time: 0.038 sec (0.036 average)

***** Running generation 28 *****

Population's average fitness: 2.52759 stdev: 0.49938
Best fitness: 3.00000 - size: (6, 6) - species 6 - id 3684
Average adjusted fitness: 0.760
Mean genetic distance 2.680, standard deviation 1.105
Population of 149 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	28	28	3.0	0.773	1
2	28	18	3.0	0.786	5
3	18	20	3.0	0.707	3
4	17	33	3.0	0.710	13
5	13	11	3.0	0.786	6
6	7	39	3.0	0.798	4

Total extinctions: 0
Generation time: 0.036 sec (0.036 average)

***** Running generation 29 *****

Population's average fitness: 2.49628 stdev: 0.52011
Best fitness: 3.00000 - size: (5, 5) - species 6 - id 4229
Average adjusted fitness: 0.708
Mean genetic distance 2.643, standard deviation 1.127
Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	29	28	3.0	0.794	0
2	29	21	3.0	0.691	6
3	19	27	3.0	0.676	4
4	18	31	3.0	0.697	0
5	14	16	2.0	0.500	7
6	8	27	3.0	0.892	5

Total extinctions: 0
Generation time: 0.038 sec (0.037 average)

***** Running generation 30 *****

Population's average fitness: 2.46572 stdev: 0.49962
Best fitness: 3.00000 - size: (4, 3) - species 6 - id 4127
Average adjusted fitness: 0.710
Mean genetic distance 2.695, standard deviation 1.092
Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	30	42	3.0	0.700	1
2	30	22	3.0	0.653	7
3	20	26	3.0	0.732	0

4	19	28	3.0	0.745	1
5	15	19	2.5	0.509	8
6	9	13	3.0	0.920	6

Total extinctions: 0

Generation time: 0.035 sec (0.037 average)

***** Running generation 31 *****

Population's average fitness: 2.44071 stdev: 0.49466

Best fitness: 3.00000 - size: (4, 3) - species 6 - id 4127

Average adjusted fitness: 0.694

Mean genetic distance 2.690, standard deviation 1.068

Population of 149 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	31	17	3.0	0.787	0
2	31	22	3.0	0.675	8
3	21	22	3.0	0.604	1
4	20	35	3.0	0.666	0
5	16	28	2.7	0.516	9
6	10	25	3.0	0.916	7

Total extinctions: 0

Generation time: 0.036 sec (0.038 average)

***** Running generation 32 *****

Population's average fitness: 2.49184 stdev: 0.48765

Best fitness: 3.00000 - size: (4, 2) - species 4 - id 4526

Average adjusted fitness: 0.586

Mean genetic distance 2.848, standard deviation 1.077

Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	32	18	3.0	0.708	1
2	32	26	3.0	0.632	9
3	22	22	3.0	0.473	2
4	21	24	3.0	0.659	1
5	17	24	3.0	0.275	10
6	11	36	3.0	0.769	8

Total extinctions: 0

Generation time: 0.036 sec (0.038 average)

***** Running generation 33 *****

Population's average fitness: 2.55615 stdev: 0.48245

Best fitness: 3.00000 - size: (6, 6) - species 6 - id 4649

Average adjusted fitness: 0.758

Mean genetic distance 2.917, standard deviation 1.004

Population of 151 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	33	14	3.0	0.715	0
2	33	21	3.0	0.815	10
3	23	23	3.0	0.683	3
4	22	18	3.0	0.796	2
5	18	20	3.0	0.634	11
6	12	24	3.0	0.903	9
7	0	31	--	--	0

Total extinctions: 0

Generation time: 0.035 sec (0.037 average)

***** Running generation 34 *****

Population's average fitness: 2.50932 stdev: 0.48909

Best fitness: 3.00000 - size: (4, 3) - species 6 - id 4657

Average adjusted fitness: 0.715

Mean genetic distance 3.107, standard deviation 1.059

Population of 150 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	34	22	3.0	0.495	1
2	34	13	3.0	0.683	11
3	24	24	3.0	0.654	4
4	23	22	3.0	0.817	3
5	19	22	3.0	0.705	12
6	13	21	3.0	0.862	10
7	1	26	3.0	0.787	0

Total extinctions: 0

Generation time: 0.039 sec (0.037 average)

***** Running generation 35 *****

Population's average fitness: 2.47731 stdev: 0.49970

Best fitness: 3.00000 - size: (4, 3) - species 6 - id 4127

Average adjusted fitness: 0.727

Mean genetic distance 3.018, standard deviation 1.116

Population of 150 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	35	27	3.0	0.685	2
2	35	15	3.0	0.728	12
3	25	22	3.0	0.593	5
4	24	23	3.0	0.790	4
5	20	21	3.0	0.703	13
6	14	21	3.0	0.705	11
7	2	21	3.0	0.888	1

Total extinctions: 0
Generation time: 0.039 sec (0.037 average)

***** Running generation 36 *****

Population's average fitness: 2.52612 stdev: 0.49472
Best fitness: 3.00000 - size: (4, 3) - species 6 - id 4127
Average adjusted fitness: 0.760
Mean genetic distance 3.009, standard deviation 1.173
Population of 150 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	36	33	3.0	0.734	3
2	36	15	3.0	0.676	13
3	26	23	3.0	0.706	6
4	25	23	3.0	0.840	5
5	21	26	3.0	0.830	14
6	15	14	3.0	0.714	12
7	3	16	3.0	0.820	2

Total extinctions: 0
Generation time: 0.037 sec (0.037 average)

***** Running generation 37 *****

Population's average fitness: 2.48106 stdev: 0.49469
Best fitness: 3.00000 - size: (4, 3) - species 6 - id 4127
Average adjusted fitness: 0.711
Mean genetic distance 2.948, standard deviation 1.156
Population of 150 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	37	29	3.0	0.665	4
2	37	17	3.0	0.661	14
3	27	23	3.0	0.670	7
4	26	20	3.0	0.584	6
5	22	26	3.0	0.785	15
6	16	14	3.0	0.795	13
7	4	21	3.0	0.820	3

Total extinctions: 0
Generation time: 0.038 sec (0.037 average)

***** Running generation 38 *****

Population's average fitness: 2.49402 stdev: 0.51052
Best fitness: 3.00000 - size: (4, 3) - species 5 - id 5341
Average adjusted fitness: 0.752
Mean genetic distance 2.890, standard deviation 1.115
Population of 150 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	38	33	3.0	0.743	5
2	38	14	3.0	0.735	15
3	28	21	3.0	0.695	8
4	27	19	3.0	0.722	7
5	23	29	3.0	0.731	16
6	17	15	3.0	0.839	14
7	5	19	3.0	0.797	4

Total extinctions: 0

Generation time: 0.037 sec (0.037 average)

***** Running generation 39 *****

Population's average fitness: 2.55209 stdev: 0.51144

Best fitness: 3.00000 - size: (5, 4) - species 5 - id 5200

Average adjusted fitness: 0.778

Mean genetic distance 2.905, standard deviation 1.106

Population of 148 members in 8 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	39	39	3.0	0.691	6
2	39	16	3.0	0.825	16
3	29	21	3.0	0.759	9
4	28	10	3.0	0.795	8
5	24	24	3.0	0.862	17
6	18	13	3.0	0.700	15
7	6	19	3.0	0.816	5
8	0	6	--	--	0

Total extinctions: 0

Generation time: 0.036 sec (0.037 average)

***** Running generation 40 *****

Population's average fitness: 2.50959 stdev: 0.49222

Best fitness: 3.00000 - size: (1, 2) - species 2 - id 5706

Average adjusted fitness: 0.608

Mean genetic distance 2.965, standard deviation 1.074

Population of 150 members in 8 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	40	33	3.0	0.544	0
2	40	11	3.0	0.674	17
3	30	17	3.0	0.585	10
4	29	19	3.0	0.479	9
5	25	23	3.0	0.690	18
6	19	8	3.0	0.714	16
7	7	23	3.0	0.922	6

8 1 16 2.0 0.256 0
 Total extinctions: 0
 Generation time: 0.037 sec (0.037 average)

***** Running generation 41 *****

Population's average fitness: 2.43952 stdev: 0.52085
 Best fitness: 3.00000 - size: (1, 2) - species 4 - id 5706
 Average adjusted fitness: 0.719
 Mean genetic distance 2.991, standard deviation 1.016
 Population of 149 members in 8 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	41	25	3.0	0.655	1
2	41	53	3.0	0.544	18
3	31	20	3.0	0.705	11
4	30	2	3.0	0.663	10
5	26	11	3.0	0.869	19
6	20	15	3.0	0.892	17
7	8	14	3.0	0.828	7
8	2	9	3.0	0.593	0

Total extinctions: 0
 Generation time: 0.038 sec (0.037 average)

***** Running generation 42 *****

Population's average fitness: 2.44636 stdev: 0.51085
 Best fitness: 3.00000 - size: (5, 6) - species 2 - id 5634

Species 5 with 11 members is stagnated: removing it
 Average adjusted fitness: 0.686
 Mean genetic distance 2.644, standard deviation 0.955
 Population of 150 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	42	27	3.0	0.692	2
2	42	24	3.0	0.751	19
3	32	22	3.0	0.639	12
4	31	3	2.0	0.500	11
6	21	19	3.0	0.750	18
7	9	14	3.0	0.860	8
8	3	41	3.0	0.611	1

Total extinctions: 0
 Generation time: 0.036 sec (0.037 average)

***** Running generation 43 *****

Population's average fitness: 2.47669 stdev: 0.49230

Best fitness: 3.00000 - size: (5, 5) - species 8 - id 5633

Species 2 with 24 members is stagnated: removing it

Average adjusted fitness: 0.626

Mean genetic distance 2.782, standard deviation 0.983

Population of 151 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	43	27	3.0	0.511	3
3	33	26	3.0	0.580	13
4	32	9	2.3	0.376	12
6	22	26	3.0	0.718	19
7	10	24	3.0	0.812	9
8	4	38	3.0	0.762	2
9	0	1	--	--	0

Total extinctions: 0

Generation time: 0.037 sec (0.037 average)

***** Running generation 44 *****

Population's average fitness: 2.48957 stdev: 0.49113

Best fitness: 3.00000 - size: (5, 5) - species 8 - id 5633

Species 6 with 26 members is stagnated: removing it

Average adjusted fitness: 0.752

Mean genetic distance 2.800, standard deviation 0.999

Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	44	11	3.0	0.760	4
3	34	25	3.0	0.624	14
4	33	34	2.5	0.533	13
7	11	28	3.0	0.929	10
8	5	33	3.0	0.748	3
9	1	19	2.8	0.916	0

Total extinctions: 0

Generation time: 0.036 sec (0.037 average)

***** Running generation 45 *****

Population's average fitness: 2.57401 stdev: 0.47093

Best fitness: 3.00000 - size: (4, 5) - species 4 - id 6377

Average adjusted fitness: 0.767

Mean genetic distance 2.859, standard deviation 1.016

Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	45	17	2.9	0.656	5

3	35	26	3.0	0.810	15
4	34	29	3.0	0.803	14
7	12	27	3.0	0.804	11
8	6	30	3.0	0.833	4
9	2	21	2.8	0.698	1

Total extinctions: 0

Generation time: 0.035 sec (0.037 average)

***** Running generation 46 *****

Population's average fitness: 2.57420 stdev: 0.45461

Best fitness: 3.00000 - size: (4, 5) - species 4 - id 6377

Average adjusted fitness: 0.578

Mean genetic distance 3.075, standard deviation 0.990

Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	46	18	3.0	0.523	6
3	36	23	3.0	0.473	16
4	35	34	3.0	0.773	15
7	13	30	3.0	0.789	12
8	7	26	3.0	0.512	5
9	3	19	3.0	0.395	0

Total extinctions: 0

Generation time: 0.035 sec (0.037 average)

***** Running generation 47 *****

Population's average fitness: 2.50071 stdev: 0.47332

Best fitness: 3.00000 - size: (4, 5) - species 4 - id 6377

Average adjusted fitness: 0.648

Mean genetic distance 3.218, standard deviation 1.082

Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	47	20	3.0	0.515	7
3	37	22	3.0	0.587	0
4	36	32	3.0	0.703	16
7	14	29	3.0	0.727	13
8	8	25	3.0	0.721	6
9	4	22	3.0	0.635	1

Total extinctions: 0

Generation time: 0.039 sec (0.037 average)

***** Running generation 48 *****

Population's average fitness: 2.62731 stdev: 0.47005

Best fitness: 3.00000 - size: (4, 5) - species 4 - id 6657

Average adjusted fitness: 0.777

Mean genetic distance 3.222, standard deviation 1.072

Population of 152 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	48	20	3.0	0.652	8
3	38	24	3.0	0.784	1
4	37	31	3.0	0.813	17
7	15	17	3.0	0.823	14
8	9	31	3.0	0.886	7
9	5	24	3.0	0.703	2
10	0	5	--	--	0

Total extinctions: 0

Generation time: 0.040 sec (0.037 average)

***** Running generation 49 *****

Population's average fitness: 2.49049 stdev: 0.48203

Best fitness: 3.00000 - size: (4, 5) - species 4 - id 6657

Average adjusted fitness: 0.724

Mean genetic distance 3.264, standard deviation 1.034

Population of 151 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	49	22	3.0	0.577	9
3	39	21	3.0	0.613	2
4	38	26	3.0	0.829	18
7	16	15	3.0	0.691	15
8	10	27	3.0	0.812	8
9	6	24	3.0	0.659	0
10	1	16	3.0	0.889	0

Total extinctions: 0

Generation time: 0.042 sec (0.037 average)

***** Running generation 50 *****

Population's average fitness: 2.53834 stdev: 0.47402

Best fitness: 3.00000 - size: (4, 3) - species 4 - id 6957

Average adjusted fitness: 0.734

Mean genetic distance 3.204, standard deviation 0.992

Population of 151 members in 8 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	50	22	3.0	0.645	10
3	40	20	3.0	0.590	3
4	39	24	3.0	0.817	19
7	17	19	3.0	0.849	16
8	11	24	3.0	0.771	9

9	7	20	3.0	0.673	0
10	2	20	3.0	0.792	1
11	0	2	--	--	0

Total extinctions: 0
Generation time: 0.094 sec (0.043 average)

***** Running generation 51 *****

Population's average fitness: 2.56290 stdev: 0.49562
Best fitness: 3.00000 - size: (4, 3) - species 4 - id 6957

Species 4 with 24 members is stagnated: removing it
Average adjusted fitness: 0.690
Mean genetic distance 3.256, standard deviation 1.021
Population of 150 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	51	23	3.0	0.639	11
3	41	20	3.0	0.555	4
7	18	25	3.0	0.823	17
8	12	24	3.0	0.810	10
9	8	21	3.0	0.664	0
10	3	26	3.0	0.907	2
11	1	11	2.0	0.429	0

Total extinctions: 0
Generation time: 0.038 sec (0.043 average)

***** Running generation 52 *****

Population's average fitness: 2.49951 stdev: 0.46476
Best fitness: 3.00000 - size: (3, 5) - species 3 - id 7340
Average adjusted fitness: 0.684
Mean genetic distance 3.216, standard deviation 0.975
Population of 151 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	52	22	3.0	0.679	12
3	42	21	3.0	0.646	0
7	19	28	3.0	0.718	18
8	13	24	3.0	0.835	11
9	9	23	3.0	0.716	1
10	4	22	3.0	0.757	3
11	2	11	2.5	0.439	0

Total extinctions: 0
Generation time: 0.041 sec (0.044 average)

***** Running generation 53 *****

Population's average fitness: 2.51773 stdev: 0.46361
 Best fitness: 3.00000 - size: (3, 5) - species 3 - id 7340
 Average adjusted fitness: 0.480
 Mean genetic distance 3.238, standard deviation 0.981
 Population of 152 members in 7 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	53	20	3.0	0.424	13
3	43	22	3.0	0.483	1
7	20	24	3.0	0.509	19
8	14	24	3.0	0.792	12
9	10	25	3.0	0.535	0
10	5	29	3.0	0.577	4
11	3	8	2.5	0.041	1

Total extinctions: 0
 Generation time: 0.040 sec (0.044 average)

***** Running generation 54 *****

Population's average fitness: 2.51580 stdev: 0.49355
 Best fitness: 3.00000 - size: (3, 5) - species 3 - id 7340

Species 7 with 24 members is stagnated: removing it
 Average adjusted fitness: 0.710
 Mean genetic distance 3.015, standard deviation 0.977
 Population of 150 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	54	17	3.0	0.690	14
3	44	26	3.0	0.745	2
8	15	27	3.0	0.770	13
9	11	33	3.0	0.656	1
10	6	31	3.0	0.789	5
11	4	16	3.0	0.609	0

Total extinctions: 0
 Generation time: 0.041 sec (0.044 average)

***** Running generation 55 *****

Population's average fitness: 2.55636 stdev: 0.47561
 Best fitness: 3.00000 - size: (7, 7) - species 11 - id 7753
 Average adjusted fitness: 0.530
 Mean genetic distance 2.997, standard deviation 0.955
 Population of 151 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	55	24	3.0	0.309	15
3	45	21	3.0	0.345	3

8	16	27	3.0	0.704	14
9	12	23	3.0	0.596	0
10	7	34	3.0	0.729	6
11	5	22	3.0	0.500	0

Total extinctions: 0

Generation time: 0.041 sec (0.045 average)

***** Running generation 56 *****

Population's average fitness: 2.48101 stdev: 0.46977

Best fitness: 3.00000 - size: (7, 7) - species 11 - id 7753

Average adjusted fitness: 0.526

Mean genetic distance 3.140, standard deviation 0.995

Population of 152 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	56	22	3.0	0.426	16
3	46	22	3.0	0.463	4
8	17	30	3.0	0.646	15
9	13	22	3.0	0.424	0
10	8	32	3.0	0.662	7
11	6	24	3.0	0.534	1

Total extinctions: 0

Generation time: 0.041 sec (0.046 average)

***** Running generation 57 *****

Population's average fitness: 2.52671 stdev: 0.48009

Best fitness: 3.00000 - size: (7, 7) - species 11 - id 7753

Average adjusted fitness: 0.516

Mean genetic distance 3.125, standard deviation 0.966

Population of 151 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	57	20	3.0	0.358	17
3	47	21	3.0	0.410	5
8	18	29	3.0	0.600	16
9	14	27	3.0	0.654	1
10	9	31	3.0	0.623	8
11	7	23	3.0	0.451	2

Total extinctions: 0

Generation time: 0.040 sec (0.046 average)

***** Running generation 58 *****

Population's average fitness: 2.46222 stdev: 0.47562

Best fitness: 3.00000 - size: (7, 7) - species 11 - id 7887

Average adjusted fitness: 0.558

Mean genetic distance 3.075, standard deviation 0.990

Population of 152 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	58	21	3.0	0.455	18
3	48	22	3.0	0.496	6
8	19	27	3.0	0.640	17
9	15	28	3.0	0.603	0
10	10	30	3.0	0.620	9
11	8	24	3.0	0.533	3

Total extinctions: 0

Generation time: 0.038 sec (0.046 average)

***** Running generation 59 *****

Population's average fitness: 2.50934 stdev: 0.47381

Best fitness: 3.00000 - size: (7, 7) - species 11 - id 7887

Average adjusted fitness: 0.492

Mean genetic distance 3.050, standard deviation 0.969

Population of 151 members in 6 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
1	59	20	3.0	0.378	19
3	49	20	3.0	0.361	7
8	20	27	3.0	0.559	18
9	16	27	3.0	0.478	0
10	11	35	3.0	0.800	10
11	9	22	3.0	0.378	4

Total extinctions: 0

Generation time: 0.039 sec (0.045 average)

***** Running generation 60 *****

Population's average fitness: 2.49289 stdev: 0.48436

Best fitness: 3.00000 - size: (7, 7) - species 11 - id 7887

Species 1 with 20 members is stagnated: removing it

Average adjusted fitness: 0.688

Mean genetic distance 2.952, standard deviation 1.015

Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	50	26	3.0	0.644	8
8	21	31	3.0	0.726	19
9	17	32	3.0	0.738	1
10	12	35	3.0	0.713	11
11	10	26	3.0	0.618	5

Total extinctions: 0

Generation time: 0.038 sec (0.040 average)

***** Running generation 61 *****

Population's average fitness: 2.56242 stdev: 0.48596

Best fitness: 3.00000 - size: (7, 7) - species 11 - id 7887

Species 8 with 31 members is stagnated: removing it

Average adjusted fitness: 0.718

Mean genetic distance 2.881, standard deviation 1.002

Population of 149 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	51	34	3.0	0.674	9
9	18	40	3.0	0.784	0
10	13	42	3.0	0.776	12
11	11	33	3.0	0.639	6

Total extinctions: 0

Generation time: 0.036 sec (0.039 average)

***** Running generation 62 *****

Population's average fitness: 2.59682 stdev: 0.48126

Best fitness: 3.00000 - size: (2, 4) - species 3 - id 8760

Average adjusted fitness: 0.587

Mean genetic distance 2.848, standard deviation 0.962

Population of 149 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	52	33	3.0	0.500	0
9	19	39	3.0	0.586	0
10	14	45	3.0	0.762	13
11	12	32	3.0	0.500	7

Total extinctions: 0

Generation time: 0.035 sec (0.039 average)

***** Running generation 63 *****

Population's average fitness: 2.47804 stdev: 0.48419

Best fitness: 3.00000 - size: (2, 4) - species 3 - id 8760

Average adjusted fitness: 0.485

Mean genetic distance 2.868, standard deviation 0.933

Population of 150 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	53	28	3.0	0.303	1
9	20	42	3.0	0.574	0
10	15	47	3.0	0.631	14

11 13 33 3.0 0.433 8
Total extinctions: 0
Generation time: 0.037 sec (0.039 average)

***** Running generation 64 *****

Population's average fitness: 2.46584 stdev: 0.48888
Best fitness: 3.00000 - size: (2, 4) - species 3 - id 8760
Average adjusted fitness: 0.500
Mean genetic distance 2.976, standard deviation 0.885
Population of 150 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	54	32	3.0	0.549	2
9	21	43	3.0	0.510	0
10	16	46	3.0	0.604	15
11	14	29	3.0	0.336	9

Total extinctions: 0
Generation time: 0.037 sec (0.038 average)

***** Running generation 65 *****

Population's average fitness: 2.55942 stdev: 0.48885
Best fitness: 3.00000 - size: (2, 4) - species 3 - id 9240
Average adjusted fitness: 0.720
Mean genetic distance 2.840, standard deviation 0.783
Population of 149 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	55	30	3.0	0.669	0
9	22	40	3.0	0.709	1
10	17	43	3.0	0.786	16
11	15	34	3.0	0.716	10
12	0	2	--	--	0

Total extinctions: 0
Generation time: 0.038 sec (0.038 average)

***** Running generation 66 *****

Population's average fitness: 2.45939 stdev: 0.50022
Best fitness: 3.00000 - size: (2, 4) - species 3 - id 9373
Average adjusted fitness: 0.641
Mean genetic distance 2.813, standard deviation 0.770
Population of 149 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	56	34	3.0	0.681	0
9	23	37	3.0	0.673	2

10	18	40	3.0	0.802	17
11	16	33	3.0	0.611	11
12	1	5	2.0	0.435	0

Total extinctions: 0

Generation time: 0.037 sec (0.037 average)

***** Running generation 67 *****

Population's average fitness: 2.48909 stdev: 0.50556

Best fitness: 3.00000 - size: (2, 4) - species 3 - id 9373

Average adjusted fitness: 0.669

Mean genetic distance 2.883, standard deviation 0.874

Population of 151 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	57	24	3.0	0.610	1
9	24	38	3.0	0.740	0
10	19	40	3.0	0.796	18
11	17	35	3.0	0.751	12
12	2	14	2.0	0.448	1

Total extinctions: 0

Generation time: 0.038 sec (0.037 average)

***** Running generation 68 *****

Population's average fitness: 2.44494 stdev: 0.49184

Best fitness: 3.00000 - size: (2, 4) - species 3 - id 9373

Average adjusted fitness: 0.377

Mean genetic distance 2.859, standard deviation 0.911

Population of 151 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	58	16	3.0	0.333	2
9	25	36	3.0	0.409	1
10	20	50	3.0	0.724	19
11	18	36	3.0	0.421	13
12	3	13	2.0	0.000	2

Total extinctions: 0

Generation time: 0.037 sec (0.037 average)

***** Running generation 69 *****

Population's average fitness: 2.39626 stdev: 0.50030

Best fitness: 3.00000 - size: (2, 4) - species 3 - id 9373

Species 10 with 50 members is stagnated: removing it

Average adjusted fitness: 0.619

Mean genetic distance 2.658, standard deviation 1.043

Population of 149 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	59	21	3.0	0.672	3
9	26	46	3.0	0.658	2
11	19	50	3.0	0.646	14
12	4	32	3.0	0.500	0

Total extinctions: 0

Generation time: 0.035 sec (0.037 average)

***** Running generation 70 *****

Population's average fitness: 2.29096 stdev: 0.47324

Best fitness: 3.00000 - size: (2, 4) - species 3 - id 9373

Average adjusted fitness: 0.648

Mean genetic distance 2.695, standard deviation 0.870

Population of 149 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	60	32	3.0	0.750	4
9	27	56	3.0	0.698	0
11	20	49	3.0	0.645	15
12	5	12	3.0	0.501	1

Total extinctions: 0

Generation time: 0.037 sec (0.037 average)

***** Running generation 71 *****

Population's average fitness: 2.28121 stdev: 0.43497

Best fitness: 3.00000 - size: (2, 4) - species 3 - id 9373

Average adjusted fitness: 0.351

Mean genetic distance 2.766, standard deviation 0.858

Population of 150 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	61	36	3.0	0.363	5
9	28	45	3.0	0.420	0
11	21	47	3.0	0.336	16
12	6	22	3.0	0.285	2

Total extinctions: 0

Generation time: 0.038 sec (0.037 average)

***** Running generation 72 *****

Population's average fitness: 2.33283 stdev: 0.47528

Best fitness: 3.00001 - size: (3, 5) - species 3 - id 10216

Average adjusted fitness: 0.660

Mean genetic distance 2.730, standard deviation 0.793

Population of 151 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	62	30	3.0	0.661	0
9	29	42	3.0	0.709	0
11	22	52	3.0	0.650	17
12	7	27	3.0	0.617	0

Total extinctions: 0

Generation time: 0.037 sec (0.037 average)

***** Running generation 73 *****

Population's average fitness: 2.27851 stdev: 0.44850

Best fitness: 3.00011 - size: (3, 5) - species 3 - id 10336

Average adjusted fitness: 0.601

Mean genetic distance 2.720, standard deviation 0.826

Population of 151 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	63	34	3.0	0.603	0
9	30	39	3.0	0.597	1
11	23	46	3.0	0.685	18
12	8	32	3.0	0.520	1

Total extinctions: 0

Generation time: 0.037 sec (0.037 average)

***** Running generation 74 *****

Population's average fitness: 2.30172 stdev: 0.44533

Best fitness: 3.00011 - size: (3, 5) - species 3 - id 10336

Average adjusted fitness: 0.304

Mean genetic distance 2.746, standard deviation 0.851

Population of 151 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	64	29	3.0	0.175	1
9	31	51	3.0	0.492	2
11	24	48	3.0	0.399	19
12	9	23	3.0	0.149	2

Total extinctions: 0

Generation time: 0.038 sec (0.037 average)

***** Running generation 75 *****

Population's average fitness: 2.39073 stdev: 0.49541

Best fitness: 3.00011 - size: (3, 5) - species 3 - id 10336

Species 11 with 48 members is stagnated: removing it

Average adjusted fitness: 0.664

Mean genetic distance 2.696, standard deviation 1.013

Population of 149 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	65	39	3.0	0.643	2
9	32	63	3.0	0.742	0
12	10	41	3.0	0.609	0
13	0	6	--	--	0

Total extinctions: 0

Generation time: 0.036 sec (0.037 average)

***** Running generation 76 *****

Population's average fitness: 2.29008 stdev: 0.45408

Best fitness: 3.03457 - size: (3, 5) - species 3 - id 10757

Average adjusted fitness: 0.581

Mean genetic distance 2.825, standard deviation 0.962

Population of 149 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	66	26	3.0	0.598	0
9	33	54	3.0	0.690	1
12	11	32	3.0	0.553	0
13	1	37	2.1	0.482	0

Total extinctions: 0

Generation time: 0.034 sec (0.037 average)

***** Running generation 77 *****

Population's average fitness: 2.27416 stdev: 0.42694

Best fitness: 3.03457 - size: (3, 5) - species 13 - id 10757

Average adjusted fitness: 0.456

Mean genetic distance 2.813, standard deviation 0.937

Population of 150 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	67	36	3.0	0.444	1
9	34	48	3.0	0.532	2
12	12	36	3.0	0.481	0
13	2	30	3.0	0.368	0

Total extinctions: 0

Generation time: 0.110 sec (0.044 average)

***** Running generation 78 *****

Population's average fitness: 2.33576 stdev: 0.45281

Best fitness: 3.08630 - size: (3, 5) - species 3 - id 11022

Average adjusted fitness: 0.296

Mean genetic distance 2.853, standard deviation 0.931

Population of 150 members in 4 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	68	35	3.1	0.235	0
9	35	51	3.0	0.428	0
12	13	36	3.0	0.295	0
13	3	28	3.0	0.225	1

Total extinctions: 0

Generation time: 0.036 sec (0.044 average)

***** Running generation 79 *****

Population's average fitness: 2.39220 stdev: 0.48043

Best fitness: 3.14938 - size: (3, 5) - species 3 - id 11211

Average adjusted fitness: 0.328

Mean genetic distance 2.813, standard deviation 0.890

Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	69	35	3.1	0.239	0
9	36	49	3.0	0.462	0
12	14	33	3.0	0.268	1
13	4	32	3.0	0.344	2
14	0	1	--	--	0

Total extinctions: 0

Generation time: 0.039 sec (0.044 average)

***** Running generation 80 *****

Population's average fitness: 2.27190 stdev: 0.43224

Best fitness: 3.14938 - size: (3, 5) - species 3 - id 11211

Average adjusted fitness: 0.495

Mean genetic distance 2.903, standard deviation 0.892

Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	70	29	3.1	0.483	1
9	37	41	3.0	0.545	1
12	15	33	3.0	0.558	2
13	5	33	3.0	0.513	3
14	1	14	2.0	0.378	0

Total extinctions: 0

Generation time: 0.040 sec (0.044 average)

***** Running generation 81 *****

Population's average fitness: 2.33808 stdev: 0.47561
 Best fitness: 3.47068 - size: (3, 6) - species 3 - id 11501
 Average adjusted fitness: 0.415
 Mean genetic distance 2.880, standard deviation 0.855
 Population of 151 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	71	28	3.5	0.446	0
9	38	41	3.0	0.531	2
12	16	35	3.0	0.437	3
13	6	34	3.0	0.391	4
14	2	13	2.0	0.271	0

Total extinctions: 0
 Generation time: 0.038 sec (0.044 average)

***** Running generation 82 *****

Population's average fitness: 2.27143 stdev: 0.45662
 Best fitness: 3.47068 - size: (3, 6) - species 3 - id 11501
 Average adjusted fitness: 0.491
 Mean genetic distance 2.922, standard deviation 0.832
 Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	72	33	3.5	0.508	1
9	39	37	3.0	0.556	0
12	17	34	3.1	0.553	0
13	7	29	3.0	0.438	5
14	3	17	2.2	0.401	0

Total extinctions: 0
 Generation time: 0.041 sec (0.045 average)

***** Running generation 83 *****

Population's average fitness: 2.27962 stdev: 0.47992
 Best fitness: 3.47821 - size: (3, 5) - species 13 - id 11728
 Average adjusted fitness: 0.506
 Mean genetic distance 2.992, standard deviation 0.945
 Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	73	33	3.5	0.523	2
9	40	34	3.0	0.531	1
12	18	33	3.1	0.531	1
13	8	30	3.5	0.533	0
14	4	20	2.2	0.414	1

Total extinctions: 0
 Generation time: 0.040 sec (0.045 average)

***** Running generation 84 *****

Population's average fitness: 2.40716 stdev: 0.50892
Best fitness: 3.70676 - size: (3, 6) - species 3 - id 11879
Average adjusted fitness: 0.513
Mean genetic distance 3.023, standard deviation 0.923
Population of 148 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	74	31	3.7	0.510	0
9	41	33	3.0	0.569	2
12	19	41	3.1	0.532	2
13	9	29	3.5	0.509	1
14	5	14	3.0	0.447	0

Total extinctions: 0
Generation time: 0.040 sec (0.045 average)

***** Running generation 85 *****

Population's average fitness: 2.29879 stdev: 0.48773
Best fitness: 3.70676 - size: (3, 6) - species 3 - id 11879
Average adjusted fitness: 0.480
Mean genetic distance 2.985, standard deviation 0.930
Population of 149 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	75	30	3.7	0.469	1
9	42	34	3.0	0.531	3
12	20	34	3.1	0.463	3
13	10	28	3.5	0.458	2
14	6	23	2.8	0.479	1

Total extinctions: 0
Generation time: 0.037 sec (0.046 average)

***** Running generation 86 *****

Population's average fitness: 2.35702 stdev: 0.49585
Best fitness: 3.70676 - size: (3, 6) - species 3 - id 11879
Average adjusted fitness: 0.501
Mean genetic distance 2.976, standard deviation 0.969
Population of 149 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	76	28	3.7	0.442	2
9	43	33	3.0	0.526	0
12	21	32	3.3	0.505	0
13	11	30	3.5	0.530	3

14 7 26 2.9 0.501 2
Total extinctions: 0
Generation time: 0.036 sec (0.046 average)

***** Running generation 87 *****

Population's average fitness: 2.33276 stdev: 0.51858
Best fitness: 3.70676 - size: (3, 6) - species 3 - id 11879
Average adjusted fitness: 0.492
Mean genetic distance 2.937, standard deviation 0.964
Population of 151 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	77	29	3.7	0.460	3
9	44	34	3.0	0.550	0
12	22	31	3.5	0.453	0
13	12	27	3.5	0.479	4
14	8	30	2.9	0.517	3

Total extinctions: 0
Generation time: 0.038 sec (0.039 average)

***** Running generation 88 *****

Population's average fitness: 2.33997 stdev: 0.47358
Best fitness: 3.70676 - size: (3, 6) - species 3 - id 11879
Average adjusted fitness: 0.493
Mean genetic distance 2.877, standard deviation 0.893
Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	78	28	3.7	0.484	4
9	45	33	3.0	0.526	1
12	23	30	3.5	0.480	1
13	13	28	3.5	0.493	5
14	9	31	2.9	0.480	4

Total extinctions: 0
Generation time: 0.036 sec (0.039 average)

***** Running generation 89 *****

Population's average fitness: 2.36151 stdev: 0.46868
Best fitness: 3.70676 - size: (3, 6) - species 3 - id 11879
Average adjusted fitness: 0.466
Mean genetic distance 2.914, standard deviation 0.910
Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	79	23	3.7	0.382	5

9	46	35	3.1	0.501	0
12	24	27	3.5	0.481	2
13	14	30	3.5	0.464	6
14	10	35	3.0	0.502	5

Total extinctions: 0

Generation time: 0.039 sec (0.039 average)

***** Running generation 90 *****

Population's average fitness: 2.38907 stdev: 0.51374

Best fitness: 3.70676 - size: (3, 6) - species 3 - id 11879

Average adjusted fitness: 0.514

Mean genetic distance 2.902, standard deviation 0.890

Population of 151 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	80	25	3.7	0.486	6
9	47	35	3.1	0.520	1
12	25	29	3.5	0.594	3
13	15	31	3.5	0.490	7
14	11	31	3.0	0.481	6

Total extinctions: 0

Generation time: 0.036 sec (0.038 average)

***** Running generation 91 *****

Population's average fitness: 2.37372 stdev: 0.49143

Best fitness: 3.70676 - size: (3, 6) - species 3 - id 11879

Average adjusted fitness: 0.483

Mean genetic distance 2.812, standard deviation 0.910

Population of 149 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	81	23	3.7	0.441	7
9	48	32	3.2	0.503	0
12	26	30	3.5	0.509	4
13	16	33	3.5	0.463	8
14	12	31	3.0	0.496	7

Total extinctions: 0

Generation time: 0.038 sec (0.038 average)

***** Running generation 92 *****

Population's average fitness: 2.41307 stdev: 0.50322

Best fitness: 3.70676 - size: (3, 6) - species 3 - id 11879

Average adjusted fitness: 0.516

Mean genetic distance 2.731, standard deviation 0.900

Population of 149 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	82	26	3.7	0.522	8
9	49	30	3.2	0.494	1
12	27	32	3.5	0.602	5
13	17	34	3.5	0.490	9
14	13	27	3.0	0.471	8

Total extinctions: 0

Generation time: 0.040 sec (0.038 average)

***** Running generation 93 *****

Population's average fitness: 2.45067 stdev: 0.50759

Best fitness: 3.70676 - size: (3, 6) - species 3 - id 11879

Average adjusted fitness: 0.315

Mean genetic distance 2.704, standard deviation 0.911

Population of 150 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	83	26	3.7	0.323	9
9	50	31	3.2	0.324	2
12	28	34	3.5	0.374	0
13	18	29	3.5	0.253	10
14	14	30	3.0	0.300	9

Total extinctions: 0

Generation time: 0.038 sec (0.038 average)

***** Running generation 94 *****

Population's average fitness: 2.39855 stdev: 0.48944

Best fitness: 3.70676 - size: (3, 6) - species 3 - id 11879

Average adjusted fitness: 0.447

Mean genetic distance 2.685, standard deviation 0.903

Population of 151 members in 5 species:

ID	age	size	fitness	adj fit	stag
====	===	====	=====	=====	====
3	84	26	3.7	0.409	10
9	51	32	3.2	0.462	3
12	29	34	3.5	0.477	1
13	19	27	3.5	0.432	11
14	15	32	3.0	0.455	10

Total extinctions: 0

Generation time: 0.036 sec (0.037 average)

***** Running generation 95 *****

Population's average fitness: 2.40218 stdev: 0.52285

Best fitness: 3.98844 - size: (3, 6) - species 13 - id 13398

Best individual in generation 95 meets fitness threshold - complexity: (3, 6)

Best genome:

Key: 13398

Fitness: 3.9884357632667

Nodes:

0 DefaultNodeGene(key=0, bias=-1.850995307942949, response=1.0, activation=sigmoid, aggregation=sum)

11 DefaultNodeGene(key=11, bias=-0.4468554040744377, response=1.0, activation=sigmoid, aggregation=sum)

2000 DefaultNodeGene(key=2000, bias=-0.8327249664965634, response=1.0, activation=sigmoid, aggregation=sum)

Connections:

DefaultConnectionGene(key=(-2, 0), weight=-7.422756441218486, enabled=True)

DefaultConnectionGene(key=(-2, 11), weight=2.941962290486626, enabled=True)

DefaultConnectionGene(key=(-1, 2000), weight=1.5009214680292384, enabled=True)

DefaultConnectionGene(key=(11, 0), weight=10.420062736104764, enabled=True)

DefaultConnectionGene(key=(2000, 0), weight=-1.9460857866804437, enabled=True)

DefaultConnectionGene(key=(2000, 11), weight=5.39334946936144, enabled=True)

Output:

input (0.0, 0.0), expected output (0.0,), got [0.10453877369977622]

input (0.0, 1.0), expected output (1.0,), got [0.9962506697598333]

input (1.0, 0.0), expected output (1.0,), got [0.9999999999999971]

input (1.0, 1.0), expected output (0.0,), got [0.024936400088971068]

Take a moment to study the evolution progress in the XOR example.

2.4 Task 3: Neat + Gym

Evolve a controller for the lunar lander starting from the above XOR and Gym examples.

The `evolve_genome` function should be changed to evaluate the fitness by computing the total reward accumulated by each network in the environment:

- for each genome create a NN from the config and the genes
- reset returns the initial observation
- at each step the action to be applied to the environment could be selected by choosing the output neuron with the highest output e.g. using the `np.argmax` function
- finally remember to set the fitness (total reward) of the genome by setting `genome.fitness`

As the lunar lander starts with different velocity at each reset the reward obtained in each simulation (episode) will be different. To obtain a more stable fitness you can average the reward over multiple

episodes. The number of episodes provide a tradeoff between fitness noise and computation time. You can set it to 5 episodes for now.

To evolve the agent with gym you can create a copy of the configuration file for the XOR example from the jupiter home and modify the number of inputs and outputs according to the lunar lander gym observation and action spaces. You can set the maximum fitness to 300 or disable the fitness termination check by setting `no_fitness_termination` to `True`. Full documentation of the parameters is available from the [neat-python doc page](#)

Before evolving the agent it would be useful to change the interface of the run function so that returns the population after the evolution and can optionally take a starting population as a parameter. In this way you can call run to evolve the population incrementally.

Evolve the controller for 10 generations and look at the behaviour of the best controller after the evolution. To do so you can use `population.best_genome` to access the best network after the 10 generations. This can then be used together with a configuration to create a network and simulate it on a new test episode calling `jrender` after each step as done previously.

You can also enable the checkpointer to save the progress every 5 generations by uncommenting the appropriate line in the XOR example. A population could be loaded from a checkpoint using `p = neat.Checkpointer().restore_checkpoint("neat-checkpoint-[NUM-CHECKPOINT]")`

```
[ ]: import neat
import numpy as np
import gym

# Define the function to evaluate genomes in the LunarLander environment
def evaluate_genomes(genomes, config):
    env = gym.make("LunarLander-v2", render_mode="rgb_array") # Use
    ↪ "rgb_array" for rendering

    for genome_id, genome in genomes:
        net = neat.nn.FeedForwardNetwork.create(genome, config)
        total_reward = 0

        # Run multiple episodes for stable fitness evaluation
        num_episodes = 5
        for _ in range(num_episodes):
            observation, _ = env.reset()
            episode_reward = 0

            while True:
                action = np.argmax(net.activate(observation)) # Choose action
                ↪ with highest output
                observation, reward, terminated, truncated, _ = env.step(action)
                episode_reward += reward

                if terminated or truncated:
                    break
```

```

        total_reward += episode_reward

        # Set genome fitness as the average reward over episodes
        genome.fitness = total_reward / num_episodes

    env.close()

# Define the run function to evolve a NEAT population
def run_neat(config_path, checkpoint=None, generations=10):
    config = neat.Config(
        neat.DefaultGenome,
        neat.DefaultReproduction,
        neat.DefaultSpeciesSet,
        neat.DefaultStagnation,
        config_path
    )

    # Load from checkpoint or create a new population
    if checkpoint:
        population = neat.Checkpointer().restore_checkpoint(checkpoint)
    else:
        population = neat.Population(config)

    # Add reporters
    population.add_reporter(neat.StdOutReporter(True))
    population.add_reporter(neat.StatisticsReporter())
    population.add_reporter(neat.Checkpointer(5)) # Save every 5 generations

    # Run evolution process
    best_genome = population.run(evaluate_genomes, generations)

    return population, best_genome

# Function to test the best controller after evolution
def test_best_controller(best_genome, config_path):
    config = neat.Config(
        neat.DefaultGenome,
        neat.DefaultReproduction,
        neat.DefaultSpeciesSet,
        neat.DefaultStagnation,
        config_path
    )

    env = gym.make("LunarLander-v2", render_mode="rgb_array")
    net = neat.nn.FeedForwardNetwork.create(best_genome, config)

    observation, _ = env.reset()

```

```

total_reward = 0
step = 0

while True:
    action = np.argmax(net.activate(observation)) # Get action from network
    observation, reward, terminated, truncated, _ = env.step(action)
    total_reward += reward
    step += 1

    if step % 5 == 0 or terminated or truncated:
        jrender(env, step, f"Total Reward: {total_reward:.2f}")

    if terminated or truncated:
        break

env.close()
print(f"Final Total Reward: {total_reward:.2f}")

# Example usage:
# config_path = "path/to/neat-config-file"
# population, best_genome = run_neat(config_path, generations=10)
# test_best_controller(best_genome, config_path)

```

[]: Ellipsis

The fitness of the agent after 10 generations should not be very high. To improve the performance it would usually require quite a high number of generations. Use the following strategies to reduce the training time:

- **Reward shaping:** Instead of evaluating the agent until the end of the episode you can stop the evaluation after 500-700 steps and decrease the fitness by 100 if the agent has not landed (terminated is False) by that time. Alternatively you can subtract a penalty for every timestep. This would bias the search toward more aggressive landing strategies, which will also terminate faster.
- **Incremental learning:** The fitness obtained averaging 5 episodes (initial conditions) is too noisy for the evolution to progress reliably, using about 20 episodes would make the fitness measurement much more reliable at the cost of substantially increasing the fitness computation time. A possible shortcut is use a fixed set of episodes for a number of generations. This can be done by passing a `seed` parameter (integer) to `env.reset` so that the fitness is evaluated over the same set of episodes across multiple generations. The episode counter could be used as the seed, potentially with some offset. In this way often 40-60 generations are usually sufficient to reach a reasonable fitness (100-200) on the training episodes. You can verify that the agent behaviour is good when tested using the seeds of the training episodes, but it can be significantly worse when the agent is tested using different seeds. Once the fitness has reached a basic flying competence on the training examples you can help the agent learn more general strategies by continuing training on either (i) different seeds, (ii) more episodes, or (iii) revert to random episodes. However there is a chance that the initial set of episodes would not be very representative of general cases and thus the evolution could overfit

to a deep local minima and take a long time to recover when presented different episodes. This can also happen if the agent is trained for too long on the same episodes.

```
[ ]: ..import neat
import numpy as np
import gym

# Define the function to evaluate genomes with reward shaping and incremental
↳ learning
def evaluate_genomes(genomes, config, num_generations, seed_offset=1000):
    env = gym.make("LunarLander-v2", render_mode="rgb_array") # Use
    ↳ "rgb_array" for rendering

    # Define evaluation settings
    max_steps = 700 # Limit evaluation to 700 steps
    penalty_per_timestep = 0.1 # Penalize per timestep to encourage faster
    ↳ landing
    num_episodes = 5 if num_generations < 20 else 20 # Use incremental learning

    for genome_id, genome in genomes:
        net = neat.nn.FeedForwardNetwork.create(genome, config)
        total_reward = 0

        for episode in range(num_episodes):
            seed = episode + seed_offset # Use fixed seed for stable evaluation
            observation, _ = env.reset(seed=seed)
            episode_reward = 0
            step = 0

            while step < max_steps:
                action = np.argmax(net.activate(observation)) # Choose action
                ↳ with highest output
                observation, reward, terminated, truncated, _ = env.step(action)
                episode_reward += reward - penalty_per_timestep # Apply
                ↳ timestep penalty
                step += 1

                if terminated or truncated:
                    break

            # Penalize if the agent didn't land after max_steps
            if not terminated:
                episode_reward -= 100

            total_reward += episode_reward

    # Set genome fitness as the average reward over episodes
```

```

        genome.fitness = total_reward / num_episodes

    env.close()

# Define the run function to evolve a NEAT population with incremental learning
def run_neat(config_path, checkpoint=None, generations=60):
    config = neat.Config(
        neat.DefaultGenome,
        neat.DefaultReproduction,
        neat.DefaultSpeciesSet,
        neat.DefaultStagnation,
        config_path
    )

    # Load from checkpoint or create a new population
    if checkpoint:
        population = neat.Checkpointer().restore_checkpoint(checkpoint)
    else:
        population = neat.Population(config)

    # Add reporters
    population.add_reporter(neat.StdOutReporter(True))
    population.add_reporter(neat.StatisticsReporter())
    population.add_reporter(neat.Checkpointer(5)) # Save every 5 generations

    # Run evolution process with adaptive learning
    best_genome = population.run(lambda g, c: evaluate_genomes(g, c,
↳generations), generations)

    return population, best_genome

# Function to test the best controller after evolution
def test_best_controller(best_genome, config_path):
    config = neat.Config(
        neat.DefaultGenome,
        neat.DefaultReproduction,
        neat.DefaultSpeciesSet,
        neat.DefaultStagnation,
        config_path
    )

    env = gym.make("LunarLander-v2", render_mode="rgb_array")
    net = neat.nn.FeedForwardNetwork.create(best_genome, config)

    observation, _ = env.reset()
    total_reward = 0
    step = 0

```

```

while True:
    action = np.argmax(net.activate(observation)) # Get action from network
    observation, reward, terminated, truncated, _ = env.step(action)
    total_reward += reward
    step += 1

    if step % 5 == 0 or terminated or truncated:
        jrender(env, step, f"Total Reward: {total_reward:.2f}")

    if terminated or truncated:
        break

env.close()
print(f"Final Total Reward: {total_reward:.2f}")

# Example usage:
# config_path = "path/to/neat-config-file"
# population, best_genome = run_neat(config_path, generations=60)
# test_best_controller(best_genome, config_path)
.

```

[]: Ellipsis