CS 747, Autumn 2022: Lecture 23

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Autumn 2022

Faculties of Human Intelligence

- Visual processing
- Speech, language processing
- Planning, problem solving
- Learning
- Communication, social interaction
- Dexterity, physical skill

Faculties of Human Intelligence

- Visual processing
- Speech, language processing
- Planning, problem solving
- Learning
- Communication, social interaction
- Dexterity, physical skill

- What enables humans to do all these things?

Faculties of Human Intelligence

- Visual processing
- Speech, language processing
- Planning, problem solving
- Learning
- Communication, social interaction
- Dexterity, physical skill

- What enables humans to do all these things?
- Why aren't other animals able to do (all) the same?

Faculties of Human Intelligence

- Visual processing
- Speech, language processing
- Planning, problem solving
- Learning
- Communication, social interaction
- Dexterity, physical skill

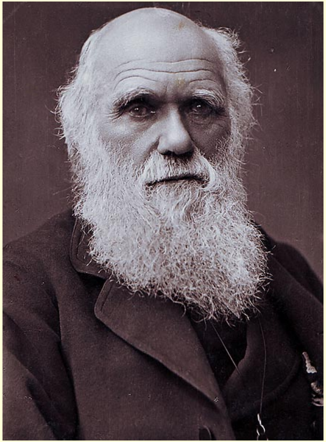
- What enables humans to do all these things?
- Why aren't other animals able to do (all) the same?
- We are born with human bodies and brains!

Faculties of Human Intelligence

- Visual processing
- Speech, language processing
- Planning, problem solving
- Learning
- Communication, social interaction
- Dexterity, physical skill

- What enables humans to do all these things?
- Why aren't other animals able to do (all) the same?
- We are born with human bodies and brains!
- And how did we get those?

Theory of Biological Evolution



Charles Darwin (1809–1882) [1]

1. https://commons.wikimedia.org/wiki/File:Charles_Darwin_photograph_by_Herbert_Rose_Barraud,_1881.jpg.

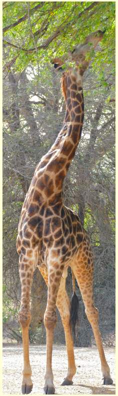
Natural Selection

[1]

1. https://commons.wikimedia.org/wiki/File:
Rothschild%27s_Giraffe_(Giraffa_camelopardalis_rothschildi)_male_(7068054987),_crop_%26_edit.jpg. CC image courtesy of Bernard DUPONT on WikiMedia Commons licensed under CC-BY-SA-2.0.

Natural Selection

[1]



[2]

1. https://commons.wikimedia.org/wiki/File:Rothschild%27s_Giraffe_(Giraffa_camelopardalis_rothschildi)_male_(7068054987),_crop_%26_edit.jpg. CC image courtesy of Bernard DUPONT on WikiMedia Commons licensed under CC-BY-SA-2.0.
2. https://commons.wikimedia.org/wiki/File:Giraffe_male_browsing_..._(33225462676).jpg CC image courtesy of Bernard DUPONT on WikiMedia Commons licensed under CC-BY-SA-2.0.

Coevolution

Angraecum sesquipedale [1]

1. `https://commons.wikimedia.org/wiki/File:Darwin%27s_Orchid_(Angraecum_sesquipedale)_(8562029223).jpg`. CC image courtesy of Bernard DUPONT on WikiMedia Commons licensed under CC-BY-SA-2.0.

Coevolution

Angraecum sesquipedale [1]

Long “spur” [2]

1. https://commons.wikimedia.org/wiki/File:Darwin%27s_Orchid_(Angraecum_sesquipedale)_(8562029223).jpg. CC image courtesy of Bernard DUPONT on WikiMedia Commons licensed under CC-BY-SA-2.0.
2. https://commons.wikimedia.org/wiki/File:Angraecum_sesquipedale_spur.jpg. CC image courtesy of Orchideen100 on WikiMedia Commons licensed under CC-BY-SA-4.0.

Coevolution

Angraecum sesquipedale [1]

Long “spur” [2]

Xanthopan morganii praedicta [3]

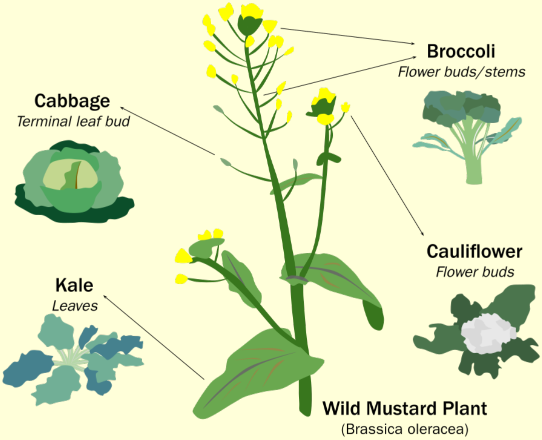
1. https://commons.wikimedia.org/wiki/File:Darwin%27s_Orchid_(Angraecum_sesquipedale)_(8562029223).jpg. CC image courtesy of Bernard DUPONT on WikiMedia Commons licensed under CC-BY-SA-2.0.
2. https://commons.wikimedia.org/wiki/File:Angraecum_sesquipedale_spur.jpg. CC image courtesy of Orchideen100 on WikiMedia Commons licensed under CC-BY-SA-4.0.
3. https://commons.wikimedia.org/wiki/File:NHM_Xanthopan_morgani.jpg. CC image courtesy of Esculapio on WikiMedia Commons licensed under CC-BY-SA-3.0.

Artificial Selection

[1]

1. https://commons.wikimedia.org/wiki/File:Dog_morphological_variation.png. CC image courtesy of Mary Bloom, American Kennel Club on WikiMedia Commons licensed under CC-BY-SA-4.0.

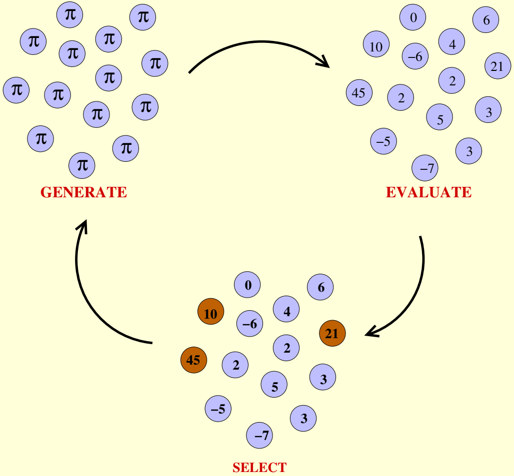
Artificial Selection



[1]

1. https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/Wild_Mustard_Plant_Selective_Breeding.svg/1000px-Wild_Mustard_Plant_Selective_Breeding.svg.png. CC image courtesy of Liwnoc on WikiMedia Commons licensed under CC-BY-SA-4.0.

Evolutionary Algorithms in Computing



- Inspired by efficiency of selection paradigm in natural world.
- Usually much less complex in terms of representation, scale, parallelisation.
- Validated by several empirical successes, although theory not very strong.
- Means for black box optimisation or policy search.

Evolution and Learning

Evolutionary Function Approximation for Reinforcement Learning.
Shimon Whiteson and Peter Stone, Journal of Machine Learning Research, 7: 877–917, 2006.

Evolution and Learning

Evolutionary Function Approximation for Reinforcement Learning.
Shimon Whiteson and Peter Stone, Journal of Machine Learning Research, 7: 877–917, 2006.

- NEAT
- NEAT+Q
- Experiments
- Discussion

Evolution and Learning

Evolutionary Function Approximation for Reinforcement Learning.
Shimon Whiteson and Peter Stone, Journal of Machine Learning Research, 7: 877–917, 2006.

- NEAT
- NEAT+Q
- Experiments
- Discussion

NeuroEvolution of Augmenting Topologies

Algorithm 2 NEAT(S, A, p, m_n, m_l, g, e)

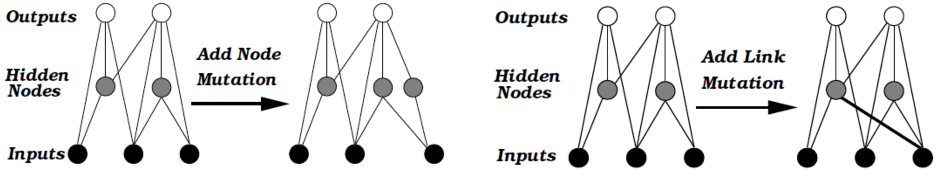
```
// S: set of all states, A: set of all actions, p: population size, m_n: node mutation rate
// m_l: link mutation rate, g: number of generations, e: episodes per generation

P[] ← INIT-POPULATION(S, A, p)              // create new population P with random networks
for i ← 1 to g do
    for j ← 1 to e do
        N, s, s' ← RANDOM(P[]), null, INIT-STATE(S)       // select a network randomly
        repeat
            Q[] ← EVAL-NET(N, s')            // evaluate selected network on current state
            a' ← argmax_i Q[i]               // select action with highest activation
            s, a ← s', a'
            r, s' ← TAKE-ACTION(a')          // take action and transition to new state
            N.fitness ← N.fitness + r        // update total reward accrued by N
        until TERMINAL-STATE?(s)
        N.episodes ← N.episodes + 1          // update total number of episodes for N
    P'[] ← new array of size p               // new array will store next generation
    for j ← 1 to p do
        P'[j] ← BREED-NET(P[])               // make a new network based on fit parents in P
        with-probability m_n: ADD-NODE-MUTATION(P'[j]) // add a node to new network
        with-probability m_l: ADD-LINK-MUTATION(P'[j]) // add a link to new network
    P[] ← P'[]
```

Evolutionary Function Approximation for Reinforcement Learning. Shimon Whiteson and Peter Stone, Journal of Machine Learning Research, 7: 877–917, 2006.

Additional Details

- Mutation by adding nodes and links.



(a) A mutation operator for adding new nodes

(b) A mutation operator for adding new links

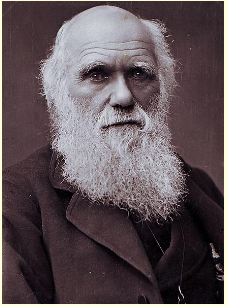
Figure 1: Examples of NEAT's mutation operators for adding structure to networks. In (a), a hidden node is added by splitting a link in two. In (b), a link, shown with a thicker black line, is added to connect two nodes.

- Crossover based on a system to track the evolution of individual genes.

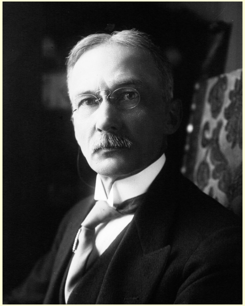
- Speciation based on explicit fitness sharing to preserve diversity in population.

Does Learning Help Evolution?

Jean-Baptiste
de Lamarck
(1744–1829)
[1]



Charles
Darwin
(1809–1882)
[2]



James Mark
Baldwin
(1861–1934)
[3]

1. https://en.wikipedia.org/wiki/Jean-Baptiste_Lamarck#/media/File:Jean-Baptiste_de_Lamarck.jpg.
2. https://commons.wikimedia.org/wiki/File:Charles_Darwin_photograph_by_Herbert_Rose_Barraud,_1881.jpg.
3. https://en.wikipedia.org/wiki/James_Mark_Baldwin#/media/File:James_Mark_Baldwin_1917.jpg.

Does Learning Help Evolution?

- In Lamarckian evolution, weight changes during an agent's lifetime get passed on to offspring.

Does Learning Help Evolution?

- In Lamarckian evolution, weight changes during an agent's lifetime get passed on to offspring.
- In Darwinian evolution, weight changes during an agent's lifetime do not get passed on to offspring.

Does Learning Help Evolution?

- In Lamarckian evolution, weight changes during an agent's lifetime get passed on to offspring.
- In Darwinian evolution, weight changes during an agent's lifetime do not get passed on to offspring.
- We now know that nature primarily implements Darwinian evolution: information flows through genes.

Does Learning Help Evolution?

- In Lamarckian evolution, weight changes during an agent's lifetime get passed on to offspring.
- In Darwinian evolution, weight changes during an agent's lifetime do not get passed on to offspring.
- We now know that nature primarily implements Darwinian evolution: information flows through genes.
- Is the Darwinian model preferable for the synthetic field of evolutionary computation, too?

Does Learning Help Evolution?

- In Lamarckian evolution, weight changes during an agent's lifetime get passed on to offspring.
- In Darwinian evolution, weight changes during an agent's lifetime do not get passed on to offspring.
- We now know that nature primarily implements Darwinian evolution: information flows through genes.
- Is the Darwinian model preferable for the synthetic field of evolutionary computation, too?
- The Baldwin effect, which examines learning in the Darwinian context, suggests that populations that learn evolve more quickly (since the starting weights only need to be approximately right, learning can "adjust" appropriately).

Does Learning Help Evolution?

- In Lamarckian evolution, weight changes during an agent's lifetime get passed on to offspring.
- In Darwinian evolution, weight changes during an agent's lifetime do not get passed on to offspring.
- We now know that nature primarily implements Darwinian evolution: information flows through genes.
- Is the Darwinian model preferable for the synthetic field of evolutionary computation, too?
- The Baldwin effect, which examines learning in the Darwinian context, suggests that populations that learn evolve more quickly (since the starting weights only need to be approximately right, learning can "adjust" appropriately).
- Over time, the starting weights themselves become more favourable (will enjoy higher fitness).

Evolution and Learning

- NEAT
- NEAT+Q
- Experiments
- Discussion

Algorithm

Algorithm 3 NEAT+Q$(S, A, c, p, m_n, m_l, g, e, \alpha, \gamma, \lambda, \varepsilon_{td})$

// *S: set of all states, A: set of all actions, c: output scale, p: population size*
// *m_n: node mutation rate, m_l: link mutation rate, g: number of generations*
// *e: number of episodes per generation, α: learning rate, γ: discount factor*
// *λ: eligibility decay rate, ε_{td}: exploration rate*

$P[] \leftarrow$ INIT-POPULATION(S, A, p) // *create new population P with random networks*
for $i \leftarrow 1$ to g **do**
  for $j \leftarrow 1$ to e **do**
    $N, s, s' \leftarrow$ RANDOM$(P[])$, null, INIT-STATE(S) // *select a network randomly*
    repeat
      $Q[] \leftarrow c \times$ EVAL-NET(N, s') // *compute value estimates for current state*

---

      with-prob(ε_{td}) $a' \leftarrow$ RANDOM(A) // *select random exploratory action*
      else $a' \leftarrow \text{argmax}_k Q[k]$ // *or select greedy action*
      if $s \neq$ null **then**
        BACKPROP$(N, s, a, (r + \gamma \max_k Q[k])/c, \alpha, \gamma, \lambda)$ // *adjust weights toward target*

      $s, a \leftarrow s', a'$
      $r, s' \leftarrow$ TAKE-ACTION(a') // *take action and transition to new state*
      $N.fitness \leftarrow N.fitness + r$ // *update total reward accrued by N*
    until TERMINAL-STATE?(s)
    $N.episodes \leftarrow N.episodes + 1$ // *update total number of episodes for N*
  $P'[] \leftarrow$ new array of size p // *new array will store next generation*
  for $j \leftarrow 1$ to p **do**
    $P'[j] \leftarrow$ BREED-NET$(P[])$ // *make a new network based on fit parents in P*
    with-probability m_n: ADD-NODE-MUTATION$(P'[j])$ // *add a node to new network*
    with-probability m_l: ADD-LINK-MUTATION$(P'[j])$ // *add a link to new network*
  $P[] \leftarrow P'[]$

Evolutionary Function Approximation for Reinforcement Learning. Shimon Whiteson and Peter Stone, Journal of Machine Learning Research, 7: 877–917, 2006.

On-line Evolutionary Computation

- By default, evolutionary computation operates in the off-line or pure exploration mode.
- To some extent the randomness in creating a population results in some exploration, and fitness-based selection amounts to exploitation.
- Yet evaluating a fixed population usually gives each member the same number of episodes.

On-line Evolutionary Computation

- By default, evolutionary computation operates in the off-line or pure exploration mode.
- To some extent the randomness in creating a population results in some exploration, and fitness-based selection amounts to exploitation.
- Yet evaluating a fixed population usually gives each member the same number of episodes.

- What if the rewards are counted on-line: that is, each fitness evaluation adds to the overall reward?

On-line Evolutionary Computation

- By default, evolutionary computation operates in the off-line or pure exploration mode.
- To some extent the randomness in creating a population results in some exploration, and fitness-based selection amounts to exploitation.
- Yet evaluating a fixed population usually gives each member the same number of episodes.

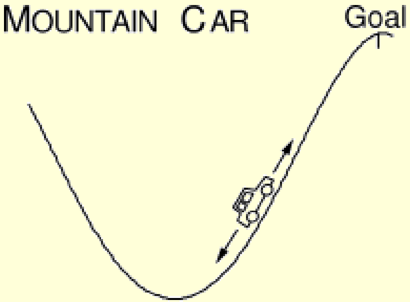
- What if the rewards are counted on-line: that is, each fitness evaluation adds to the overall reward?

- Under ϵ_{ec}-greedy selection, we pick the current best (highest empirical average of fitness) individual w.p. $1 - \epsilon_{ec}$; w.p. ϵ_{ec} we pick an individual uniformly at random.
- Under Softmax selection, we pick individual with fitness f w.p. proportional to $e^{f/\tau}$, where τ is the "temperature".

Evolution and Learning

- NEAT
- NEAT+Q
- Experiments
- Discussion

Task 1: Mountain Car



Reinforcement Learning: An Introduction. Richard S. Sutton and Andrew G. Barto, 1^{st} edition, MIT Press, 1998.

Task 2: Server Job Scheduling

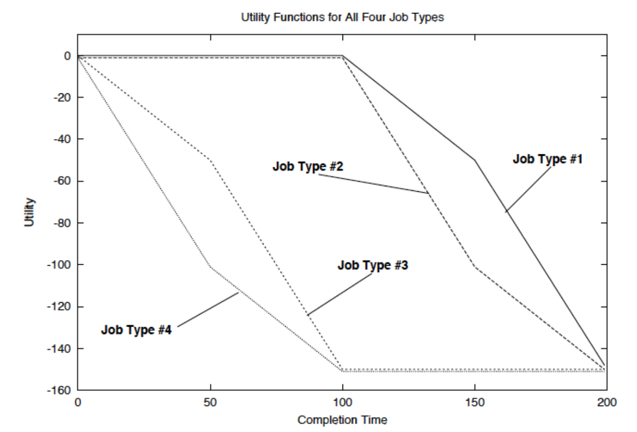


Figure 3: The four utility functions used in our experiments.

Evolutionary Function Approximation for Reinforcement Learning. Shimon Whiteson and Peter Stone, Journal of Machine Learning Research, 7: 877–917, 2006.

NEAT+Q vs. NEAT

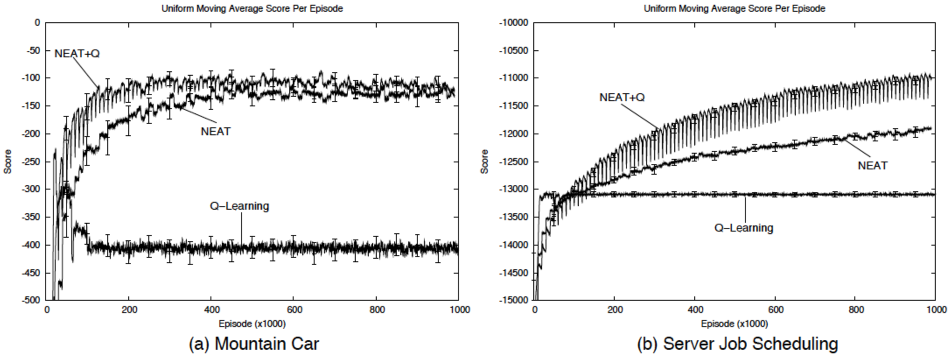


Figure 4: A comparison of the performance of manual and evolutionary function approximators in the mountain car and server job scheduling domains.

Evolutionary Function Approximation for Reinforcement Learning. Shimon Whiteson and Peter Stone, Journal of Machine Learning Research, 7: 877–917, 2006.

Topologies Evolved by NEAT+Q

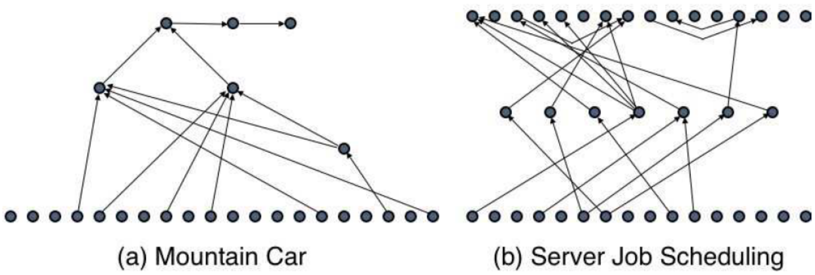


Figure 5: Typical examples of the topologies of the best networks evolved by NEAT+Q in both the mountain car and scheduling domains. Input nodes are on the bottom, hidden nodes in the middle, and output nodes on top. In addition to the links shown, each input node is directly connected to each output node. Note that two output nodes can be directly connected, in which case the activation of one node serves not only as an output of the network, but as an input to the other node.

Evolutionary Function Approximation for Reinforcement Learning. Shimon Whiteson and Peter Stone, Journal of Machine Learning Research, 7: 877–917, 2006.

On-line NEAT+Q

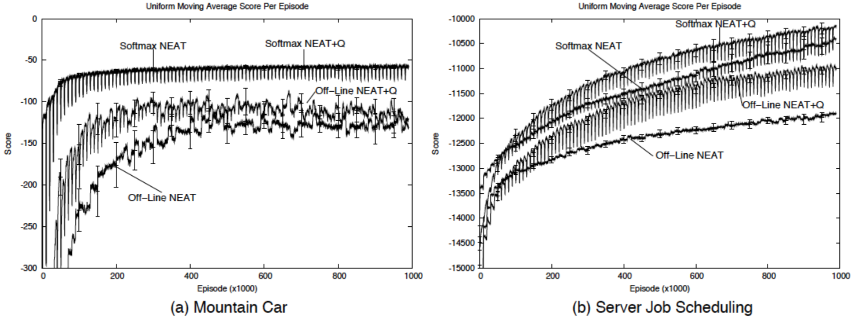


Figure 7: The performance of combining evolutionary function approximation with on-line evolutionary computation compared to using each individually in the mountain car and server job scheduling domains.

Evolutionary Function Approximation for Reinforcement Learning. Shimon Whiteson and Peter Stone, Journal of Machine Learning Research, 7: 877–917, 2006.

Darwinian vs. Lamarckian Variants

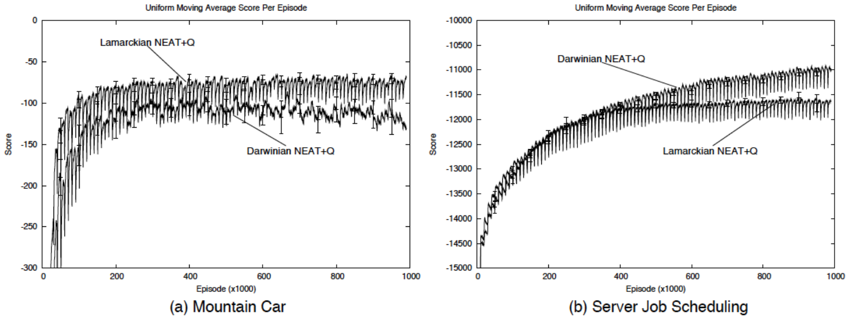


Figure 10: A comparison of Darwinian and Lamarckian NEAT+Q in the mountain car and server job scheduling domains.

Evolutionary Function Approximation for Reinforcement Learning. Shimon Whiteson and Peter Stone, Journal of Machine Learning Research, 7: 877–917, 2006.

Evolution and Learning

- NEAT
- NEAT+Q
- Experiments
- Discussion

Conclusion

- Evolution plays a primary role in animal intelligence.
- Modern ML has mostly focused on within-lifetime learning, with evolutionary computation treated as an approach for policy search.
- This week's article considers evolution as an outer loop and learning within an inner loop.
- Evolutionary computation highly parallelisable, even if it usually takes a much higher aggregate number of samples.
- Synthetic approaches need not be faithful to nature, yet there are many factors in biological evolution to be understood better and incorporated:
  - Cooperation and competition among individuals, species;
  - Implicit and explicit communication;
  - Steady-state populations.

Key References

David Ackley and Michael Littman, 1991. Interactions Between Learning and Evolution. Artificial Life II, SFI Studies in the Sciences of Complexity, 10:487–509.

Kenneth O. Stanley and Risto Miikkulainen, 2002. Evolving Neural Networks Through Augmenting Topologies. Evolutionary Computation, 2:99–127.

NEAT code: `https://www.cs.ucf.edu/~kstanley/neat.html`.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever, 2017. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. Available at: `https://arxiv.org/pdf/1703.03864.pdf`.