CAP 6635 – Artificial Intelligence

Lecture 13: How do machines learn? (Part 5)

Oge Marques, PhD

Professor

College of Engineering and Computer Science

College of Business

FLORIDA ATLANTIC UNIVERSITY

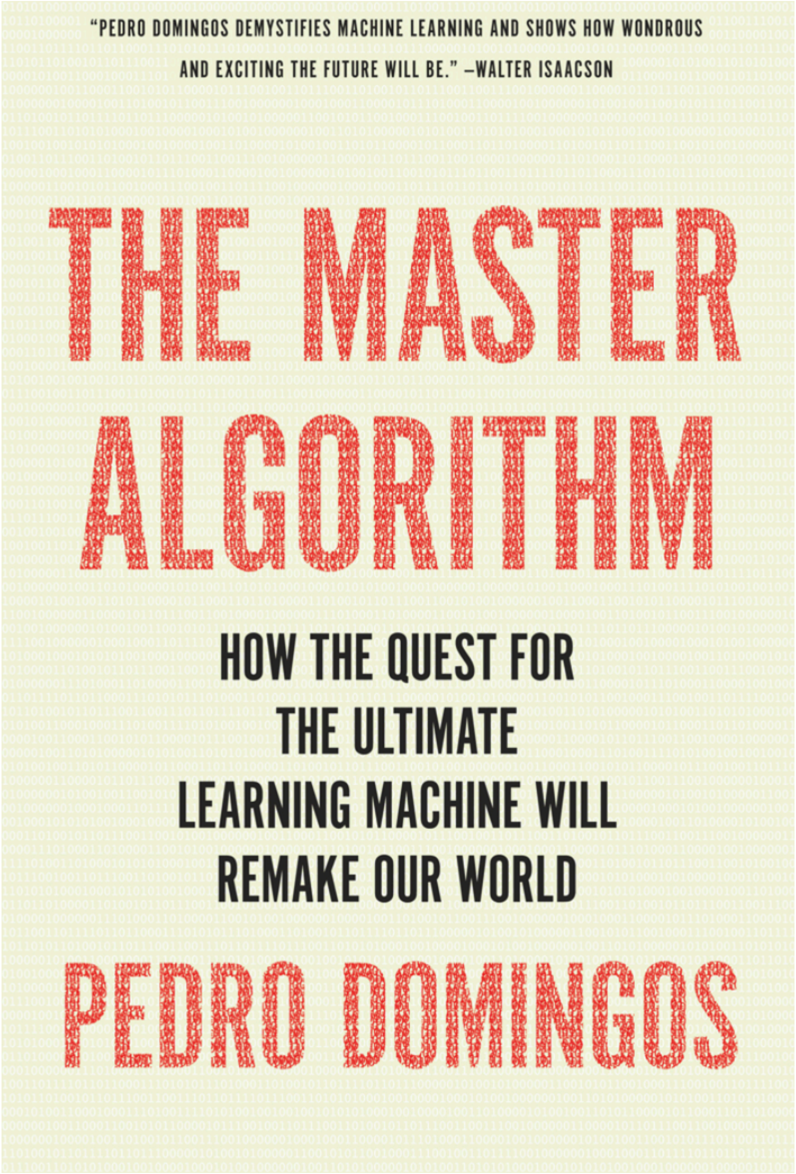
@ProfessorOge

ProfessorOgeMarques

The Master Algorithm

(by Pedro Domingos)

"All knowledge—past, present, and future—can be derived from data by a single, universal learning algorithm."



The Evolutionaries

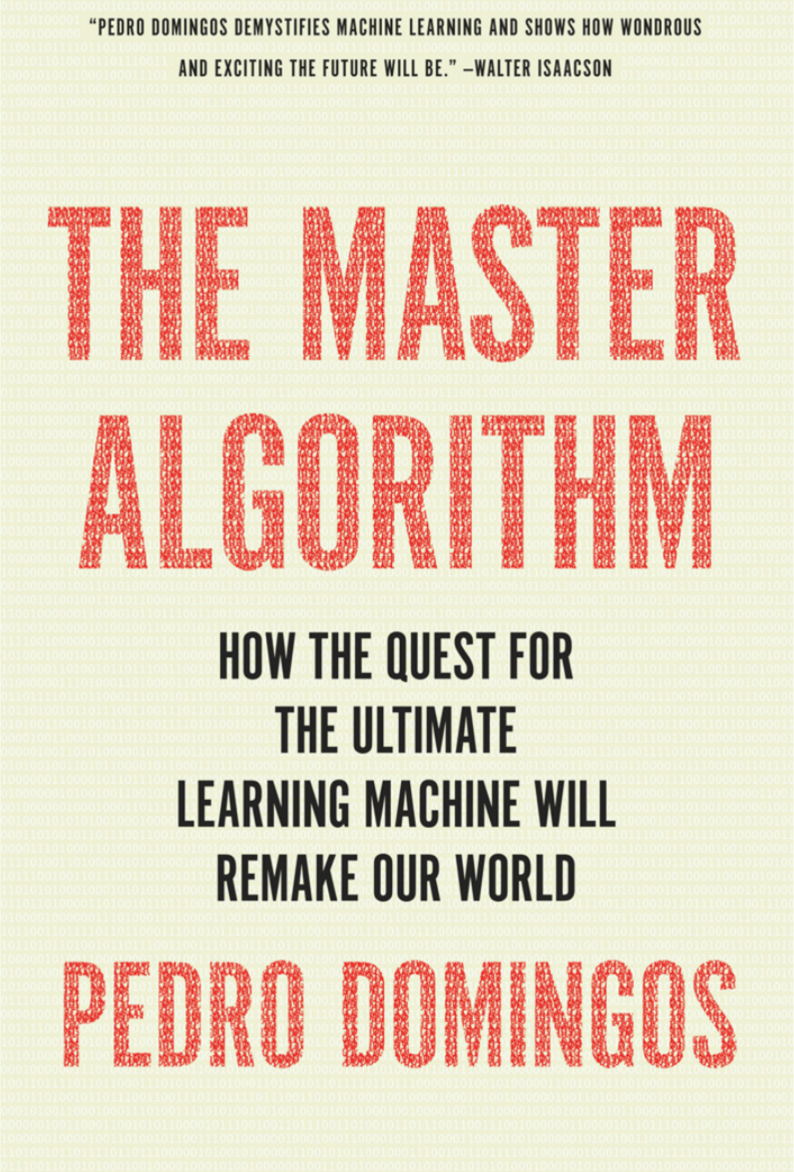
THE NATURE OF CODE

DANIEL SHIFFMAN

https://natureofcode.com/

- Chapter 9: The Evolution of Code

Back to the book...



Evolutionaries: remarks

- The exploration-exploitation dilemma
  - Genetic algorithms are much less likely than backprop to get stuck in a local optimum and in principle better able to come up with something truly new.
    - But they are also much more difficult to analyze.
  - One of the most important problems in machine learning—and life—is the exploration-exploitation dilemma.
    - If you've found something that works, should you just keep doing it?
    - Or is it better to try new things, knowing it could be a waste of time but also might lead to a better solution?
  - "A genetic algorithm is like the ringleader of a group of gamblers, playing slot machines in every casino in town at the same time." -- Pedro Domingos.

Evolutionaries: remarks

- Nurturing nature
  - Evolutionaries and connectionists have something important in common: they both design learning algorithms inspired by nature.
  - But then they part ways.
    - Evolutionaries focus on learning structure; to them, fine-tuning an evolved structure by optimizing parameters is of secondary importance.
    - In contrast, connectionists prefer to take a simple, hand-coded structure with lots of connections and let weight learning do all the work.
  - There are good arguments on both sides.

Final verdict

- “Most of all, the goal of machine learning is to find the best possible learning algorithm, by any means available, and evolution and the brain are unlikely to provide it.”

-- Pedro Domingos.
“The Master Algorithm”

- “Neuroevolution is the only branch of AI with an actual proof of concept: brains *did* evolve, so we know that's one way to produce intelligence.”

-- Kenneth O. Stanley (UCF and Uber AI Labs)

Sidebar

Deep Neuroevolution:

deep neural network training through evolutionary algorithms

Oge Marques, PhD

Oslo, Norway – May 2018

Take-home message

- The use of evolutionary algorithms for deep neural networks has the potential to evolve incrementally complex network architectures capable of solving more challenging problems on the path toward Artificial General Intelligence (AGI).
- Recently published results by leading research groups describe solutions in the field of reinforcement learning that are faster to train and produce comparable or better results than the state of the art.
- This is a very promising area that is likely to grow in size and importance in the next 5-10 years.

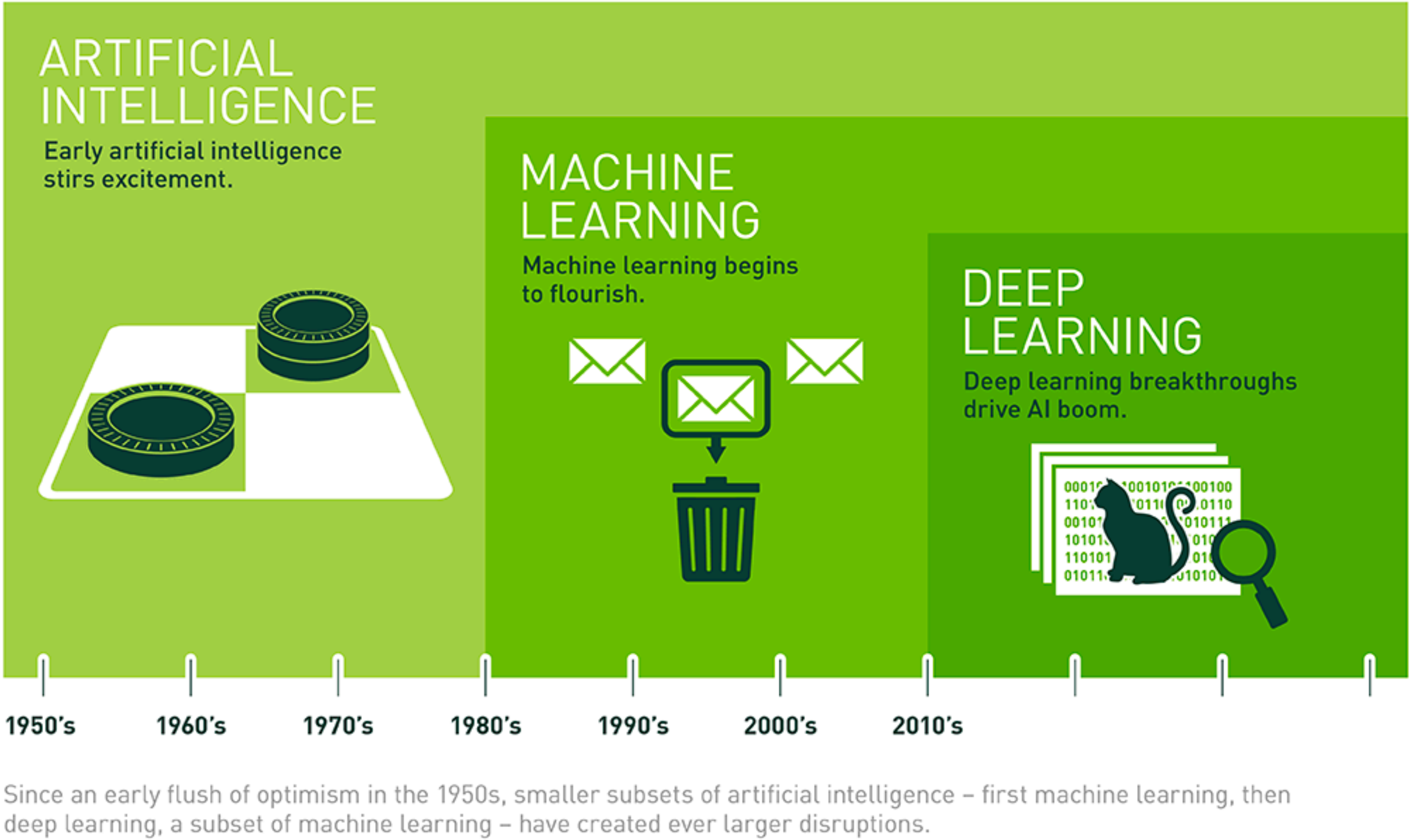
Outline

- Since ***deep neuroevolution = deep learning + evolutionary algorithms***, we will:

1. Review the basic concepts of **deep learning** and **neural networks** in general.
2. Explain the basic process behind **evolutionary algorithms**.
3. Combine (1) and (2) and introduce the topic of **neuroevolution**.
4. Introduce **Reinforcement Learning (RL)** and point to deep learning approaches to solve RL problems.
5. Highlight prominent recent examples of using **neuroevolution** on deep RL benchmarks, i.e. **deep neuroevolution**.
6. Provide resources for further reading and hands-on experimentation.

Deep Learning


DEEP LEARNING...
DEEP LEARNING EVERYWHERE
memegenerator.net



Source: NVIDIA Blog

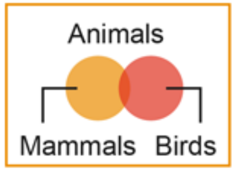
A look at
Machine learning evolution

Overview

For decades, individual “tribes” of artificial intelligence researchers have vied with one another for dominance. Is the time ripe now for tribes to collaborate? They may be forced to, as collaboration and algorithm blending are the only ways to reach true artificial general intelligence (AGI). Here’s a look back at how machine learning methods have evolved and what the future may look like.

What are the five tribes?

Symbolists



Use symbols, rules, and logic to represent knowledge and draw logical inference

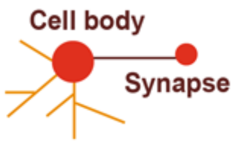
Favored algorithm
Rules and decision trees

Bayesians

Assess the likelihood of occurrence for probabilistic inference

Favored algorithm
Naive Bayes or Markov

Connectionists



Recognize and generalize patterns dynamically with matrices of probabilistic, weighted neurons

Favored algorithm
Neural networks

Evolutionaries

Generate variations and then assess the fitness of each for a given purpose

Favored algorithm
Genetic programs

Analogizers

Optimize a function in light of constraints (“going as high as you can while staying on the road”)

Favored algorithm
Support vectors

Source: Pedro Domingos, *The Master Algorithm*, 2015

Image from: http://usblogs.pwc.com/emerging-technology/machine-learning-evolution-infographic/

Phases of evolution

1980s

Predominant tribe
Symbolists

Architecture
Server or mainframe

Predominant theory
Knowledge engineering

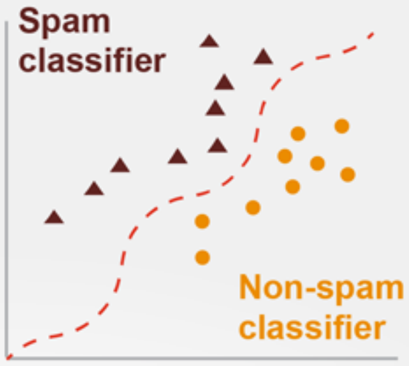
Basic decision logic:
Decision support systems with limited utility

1990s to 2000

Predominant tribe
Bayesians

Architecture
Small server clusters

Predominant theory
Probability theory



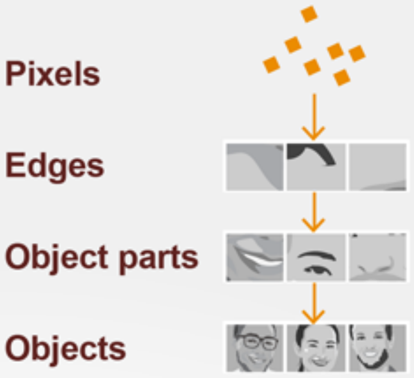
Classification:
Scalable comparison and contrast that's good enough for many purposes

Early to mid-2010s

Predominant tribe
Connectionists

Architecture
Large server farms (the cloud)

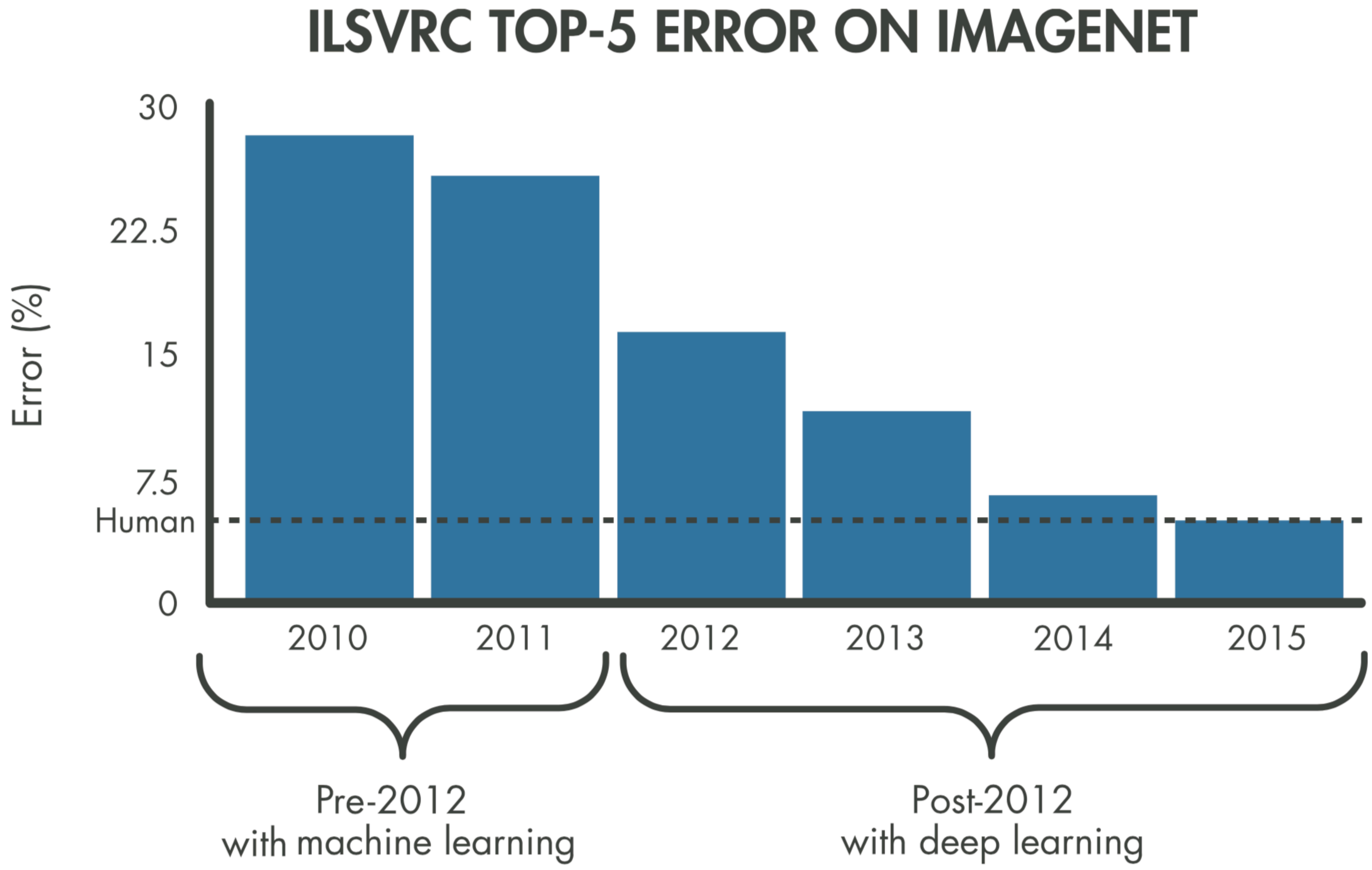
Predominant theory
Neuroscience and probability



Recognition:
More precise image and voice recognition, translation, sentiment analysis, etc.

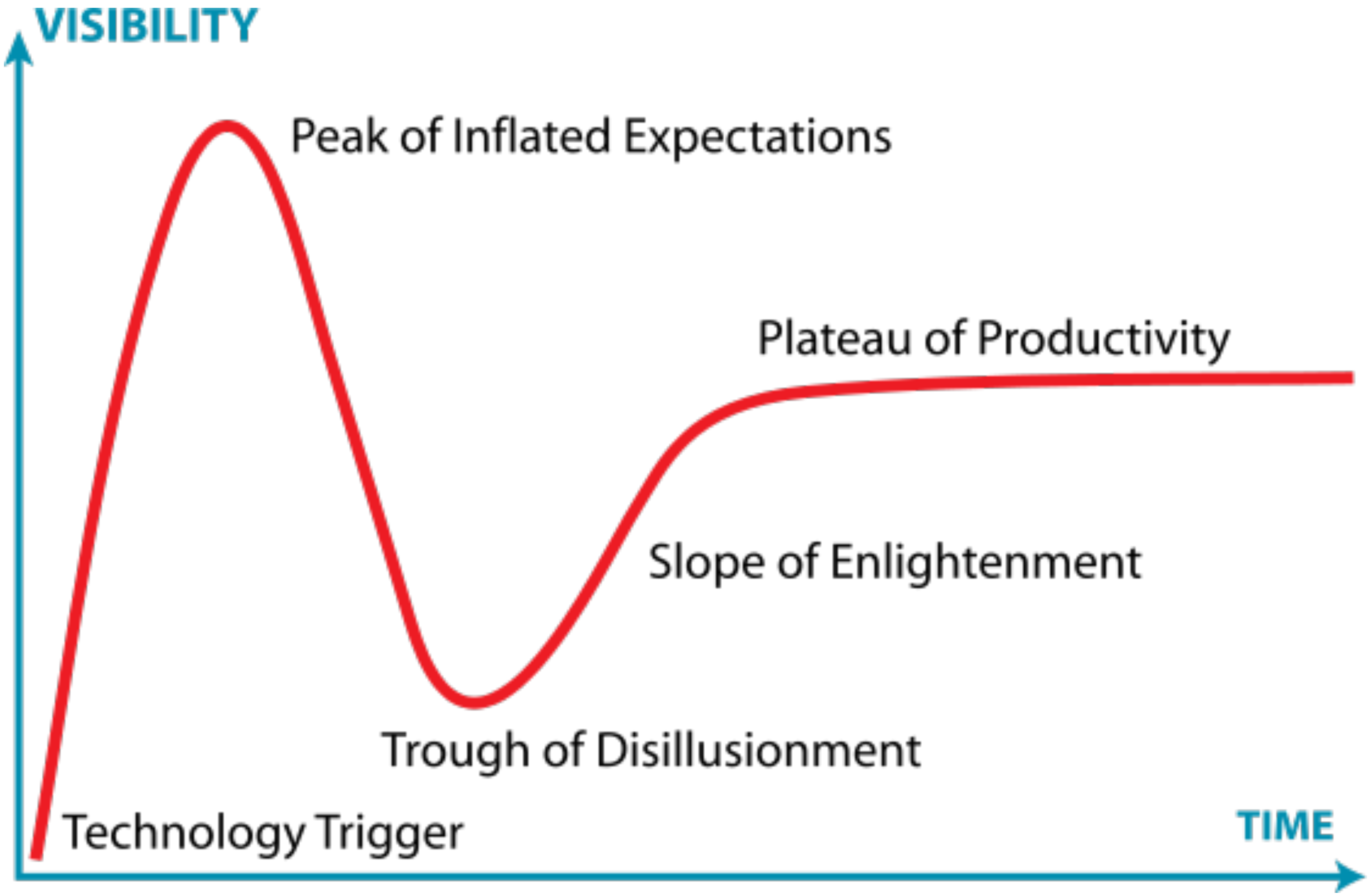
Image from: http://usblogs.pwc.com/emerging-technology/machine-learning-evolution-infographic/

Deep Learning: success and hype



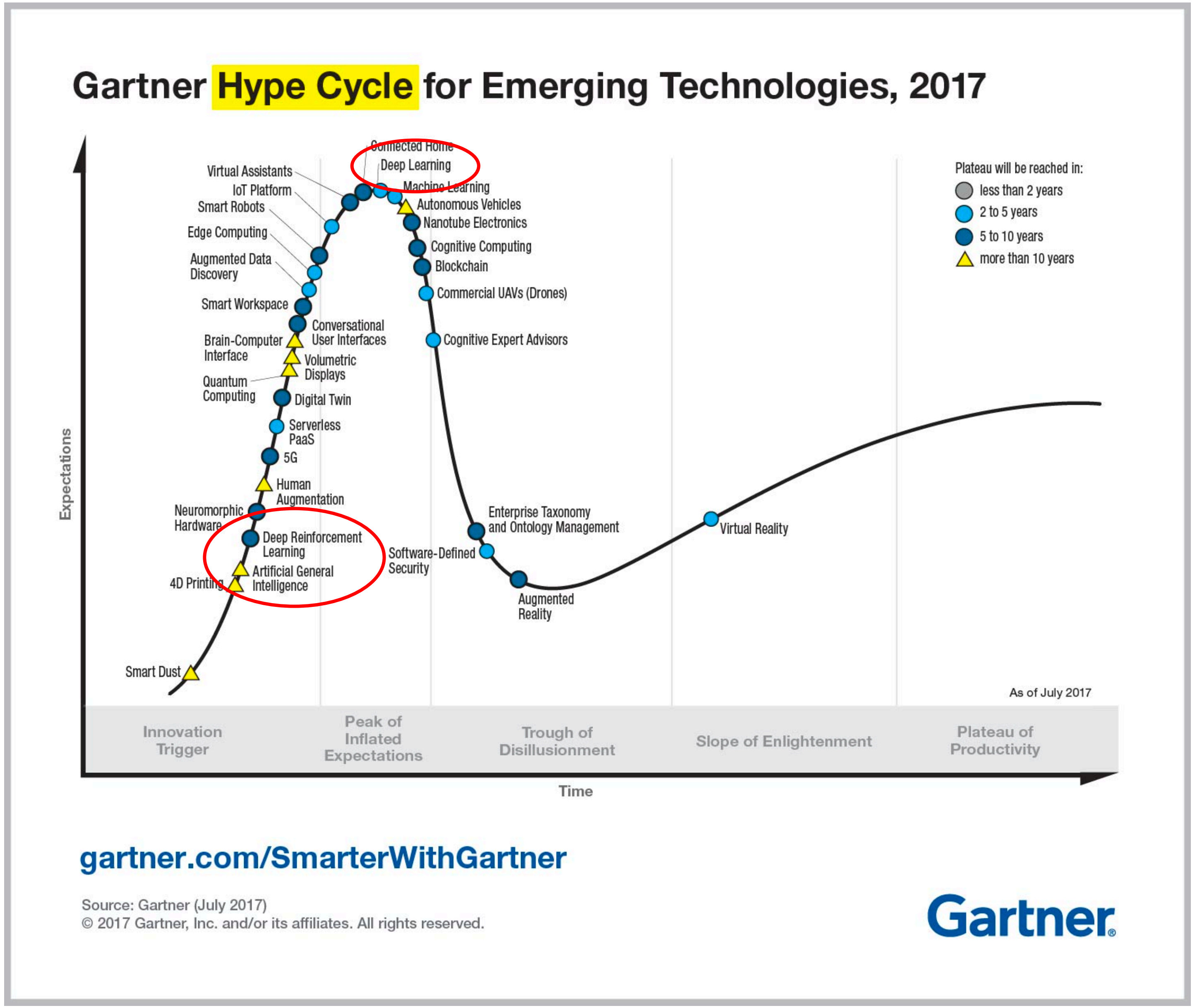
Source: (Mathworks 2017)

Deep Learning: success and hype



Source: (Wikimedia)

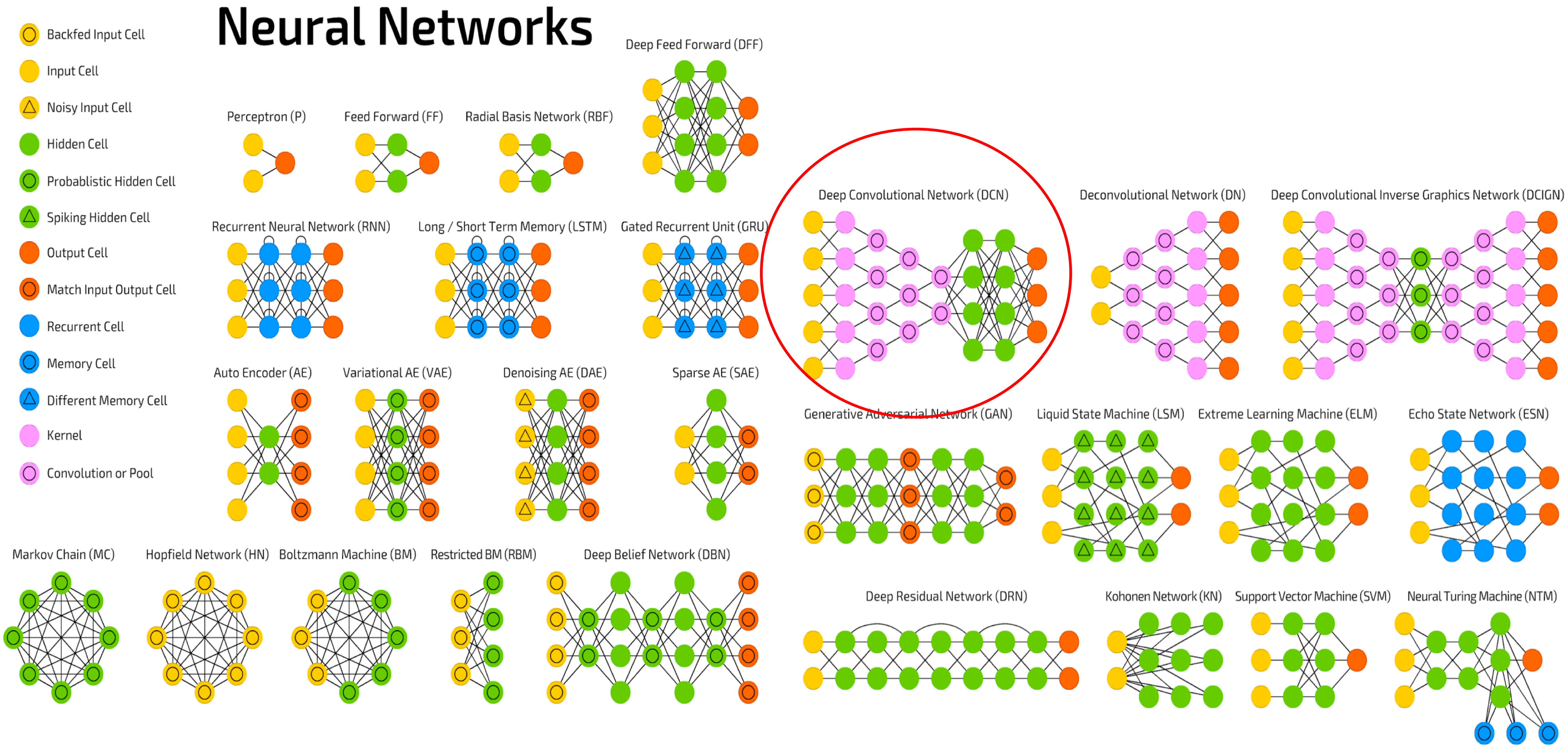
Deep Learning: success and hype



Deep Learning: limitations and criticism

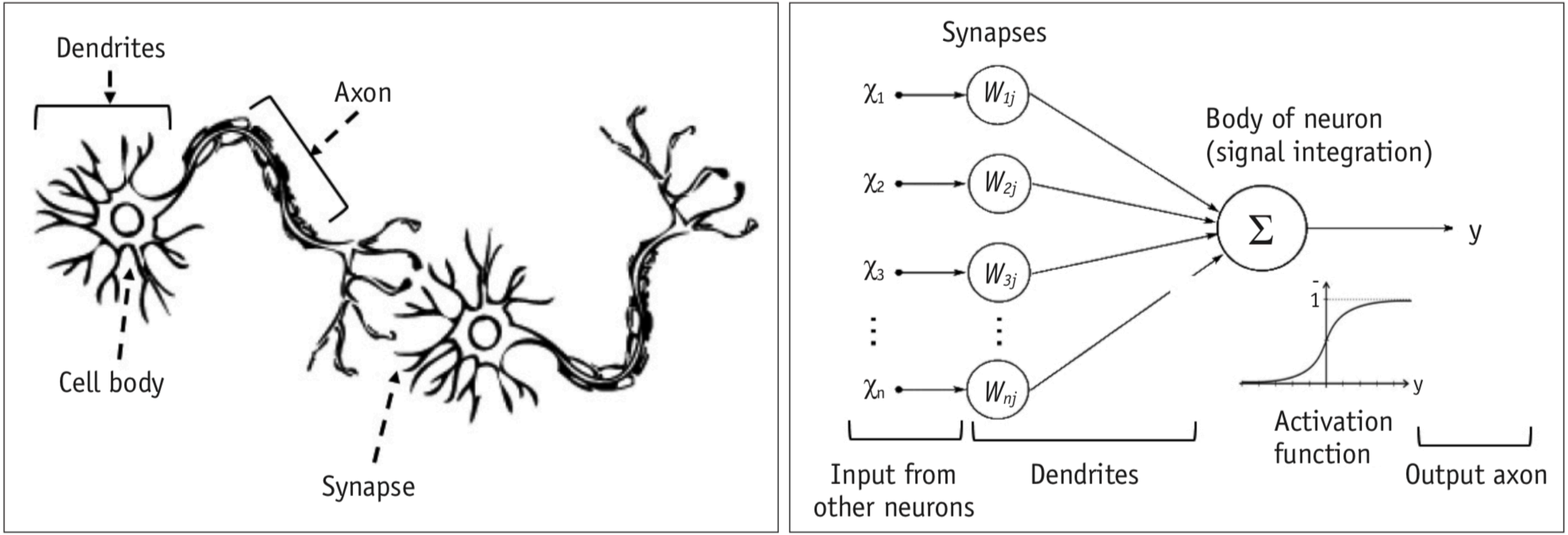
- Data hungry
- Computationally intensive
- Slow to train
- Not sufficiently transparent
- Time-consuming and ad-hoc (hyperparameter) optimization
- Problematic software development pipeline
- Fixed architectures
- … (much, much more)

Deep Learning: many flavors



Artificial Neural Networks (ANNs)

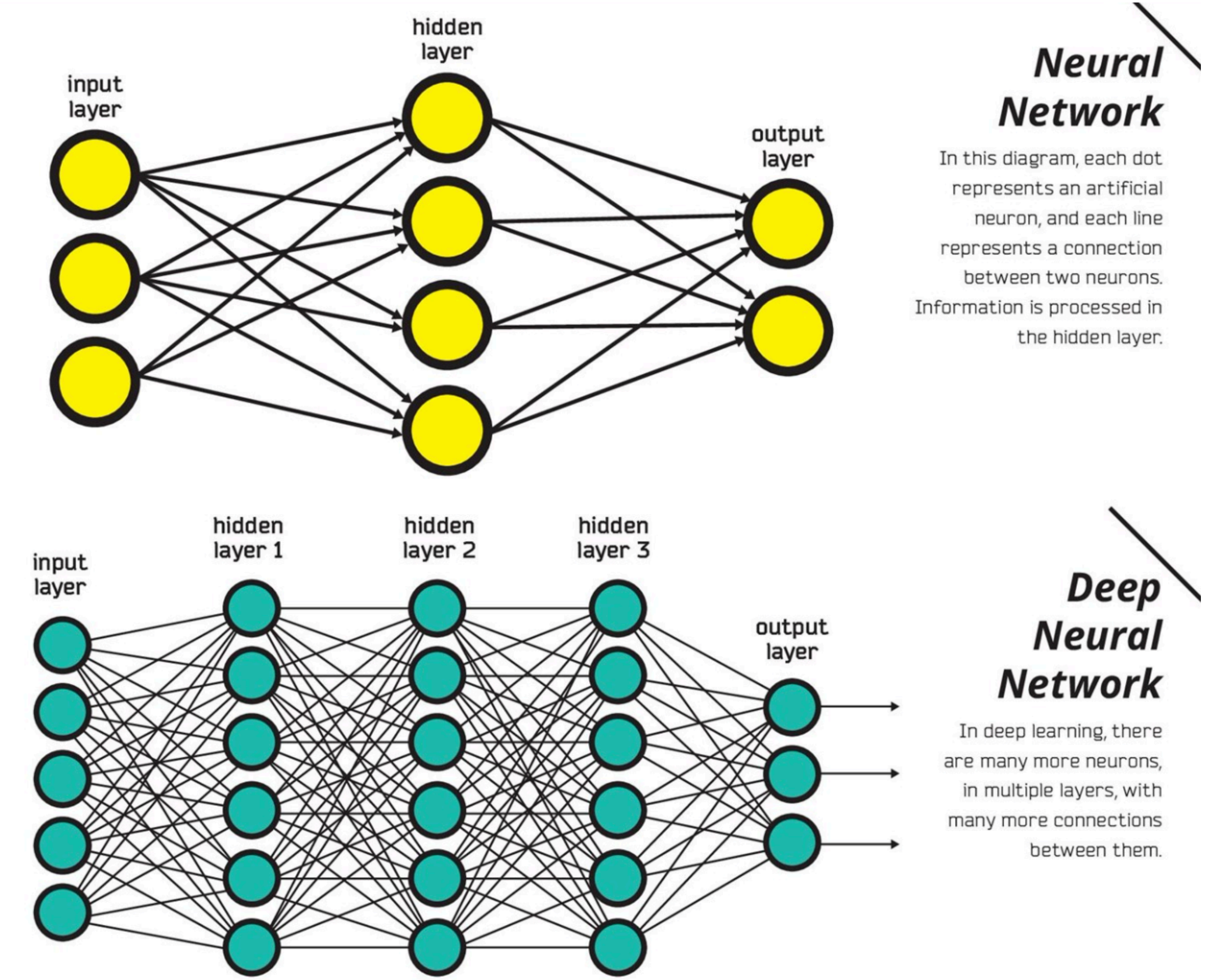
Artificial Neural Networks (ANNs): the neuron



A **B**

Korean J Radiol 18(4), Jul/Aug 2017

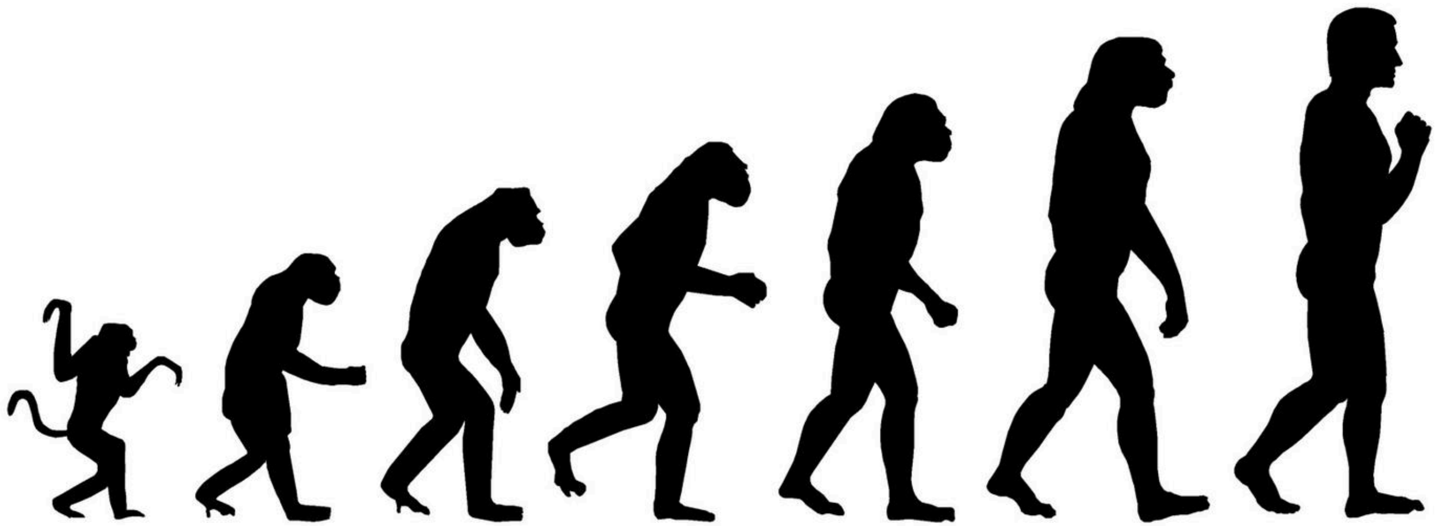
Artificial Neural Networks (ANNs)



Question:
Where did these connections come from?

- In deep neural networks (DNNs): from the mind of the human designer.

- In the human brain: the 100-trillion-connection architecture ***evolved*** through a Darwinian process over many millions of years.



Different types of learning (a partial list)

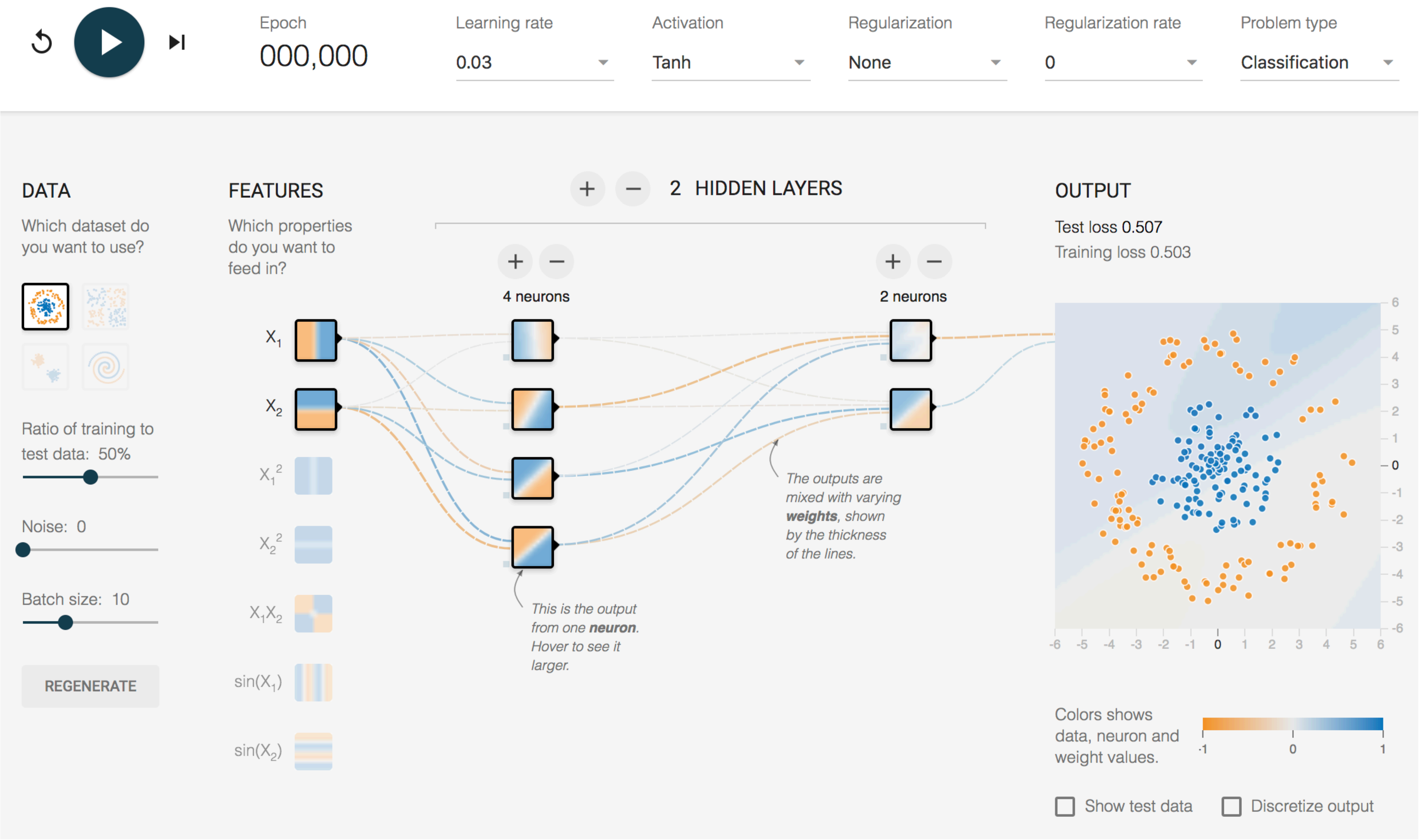
- **Supervised learning**
  - Input = data + labels
  - Goal: learn how to assign the correct label to previously unseen data
- **Unsupervised learning**
  - Input = unlabeled data
  - Goal: find patterns in the data and organize (split/cluster) accordingly.
- **Reinforcement learning**
  - A strategy based on observation:
    - Its neural network makes a decision with an *outcome* and observes its *environment.* If the observation is negative, the network can adjust its weights in order to make a different decision the next time.

How do neural networks learn?

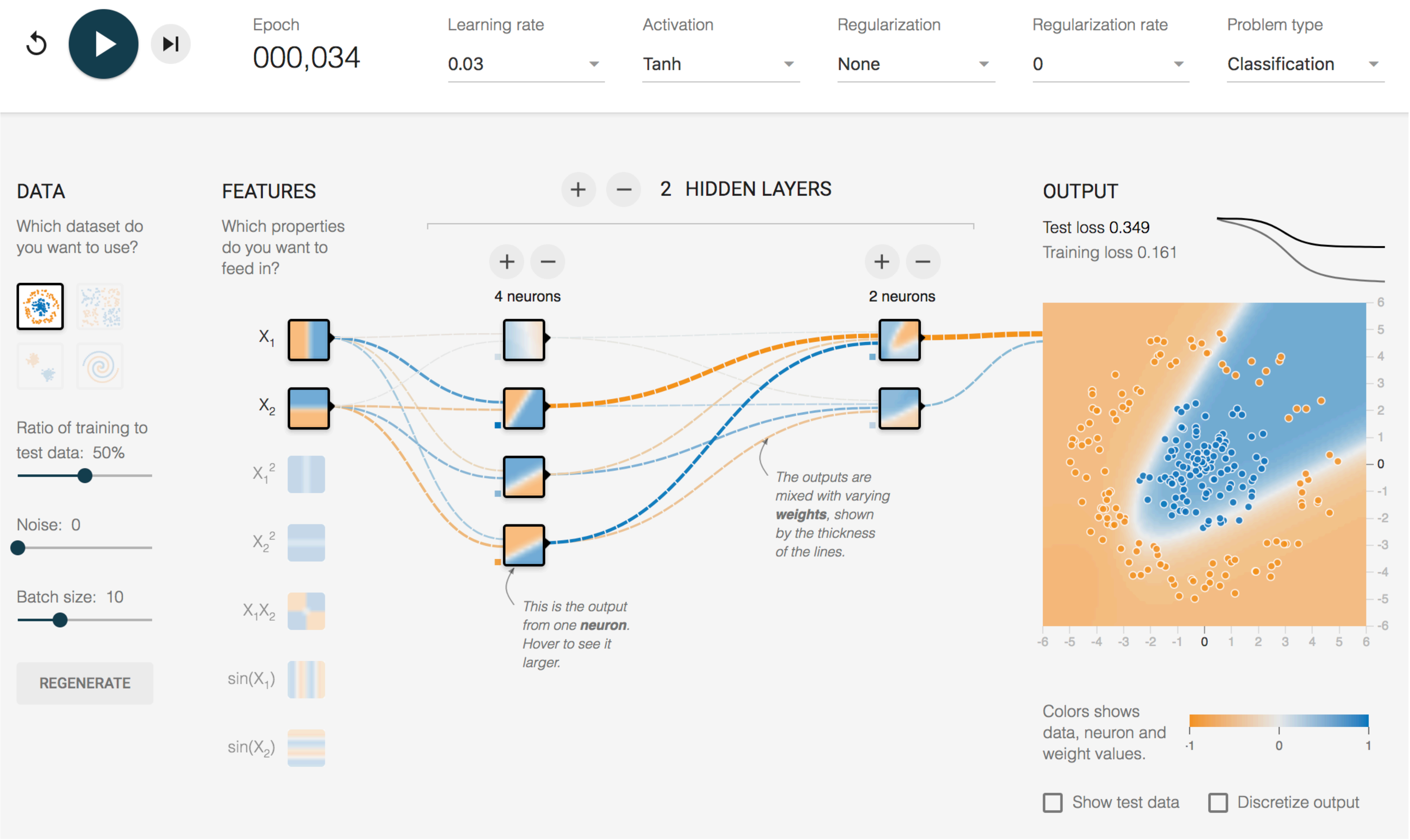
- Start with a **fixed** topology:
    - Number of layers
    - Number of neurons per layer
    - Types of layers

- During the **Training** phase:
    1. **Initialize** weights and biases with random values.
    2. **Feedforward** pass: compute network outputs.
    3. Compare network outputs against target outputs and compute **error**.
    4. **Backpropagation**: adjust weights and biases in a direction (given by the Gradient Descent algorithm) that will likely decrease the error.
    5. **Repeat** steps 2-4 for a number of **epochs**.
    6. **Freeze** weights and biases.

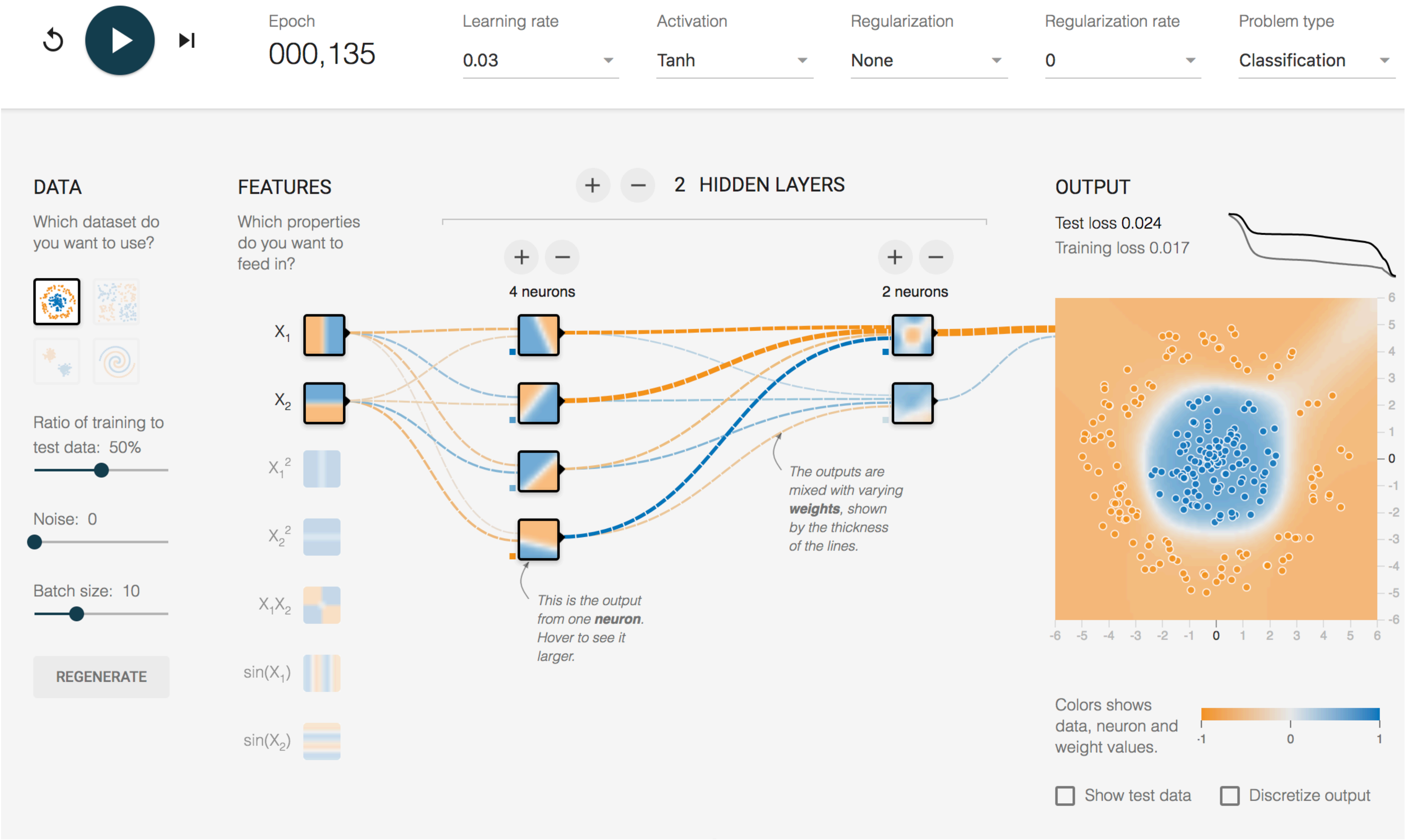
Example using TensorFlow Playground



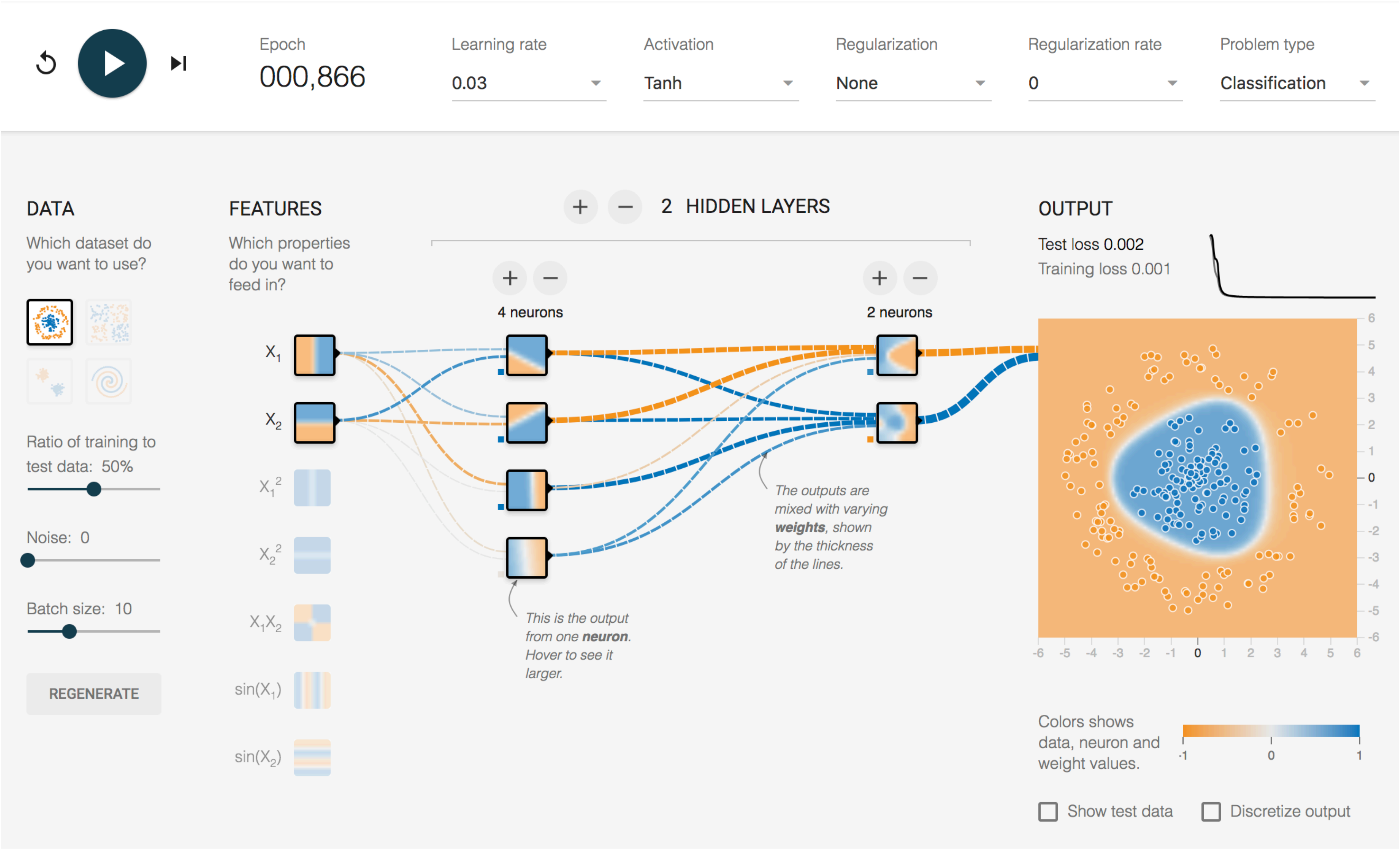
Example using TensorFlow Playground



Example using TensorFlow Playground



Example using TensorFlow Playground



TensorFlow Playground

- Go play!

https://playground.tensorflow.org/

Evolutionary Algorithms

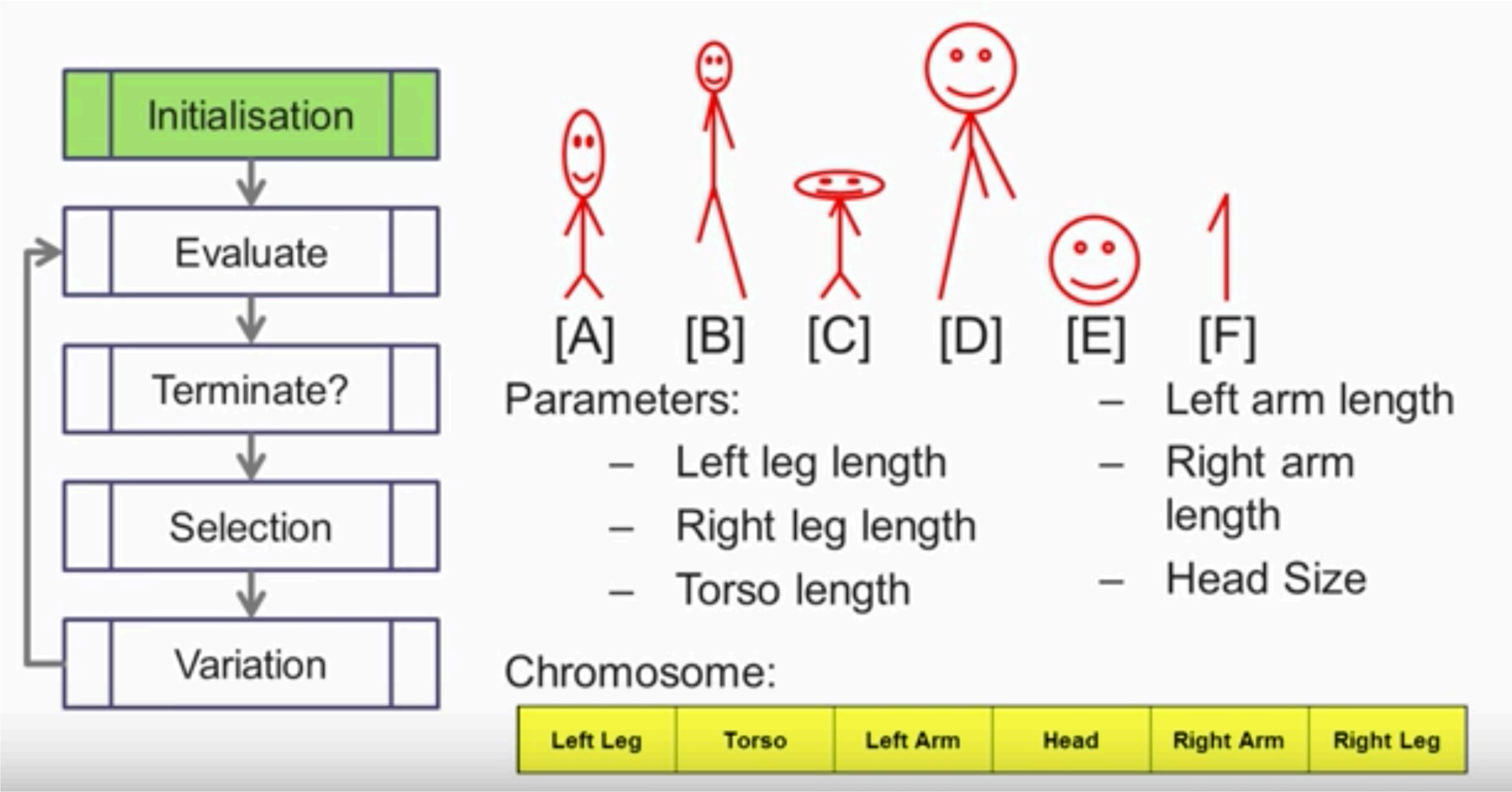
Evolutionary Algorithms: preconditions

Darwinian natural selection

- In order for natural selection to occur as it does in nature, all three of these elements must be present:
  - ***Heredity.*** There must be a process in place by which children receive the properties of their parents.
  - ***Variation.*** There must be a variety of traits present in the population or a means with which to introduce variation.
  - ***Selection.*** There must be a mechanism by which some members of a population have the opportunity to be parents and pass down their genetic information and some do not. This is typically referred to as "survival of the fittest."

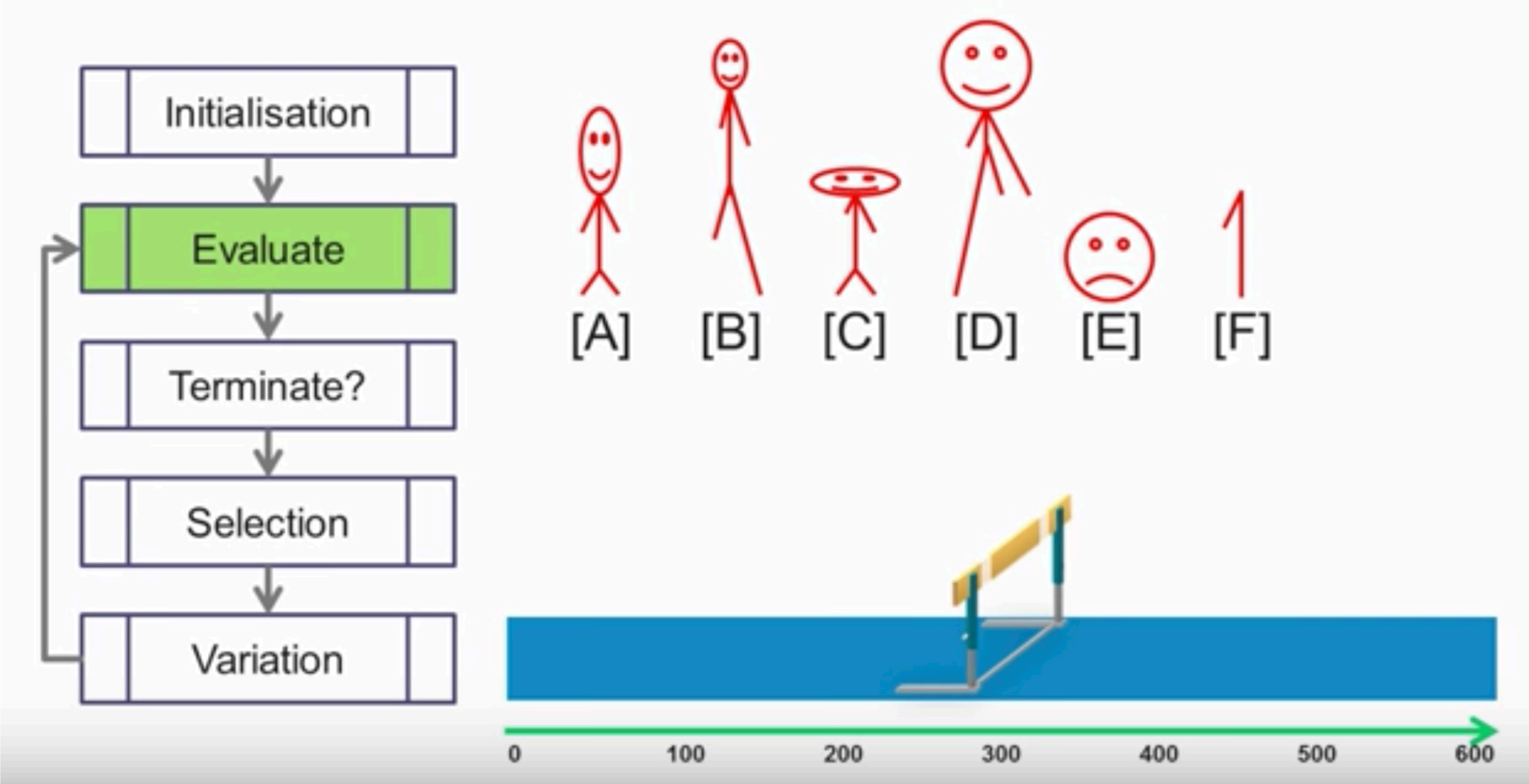
Source: http://natureofcode.com/book/chapter-9-the-evolution-of-code/

Evolutionary Algorithms: how does it work?



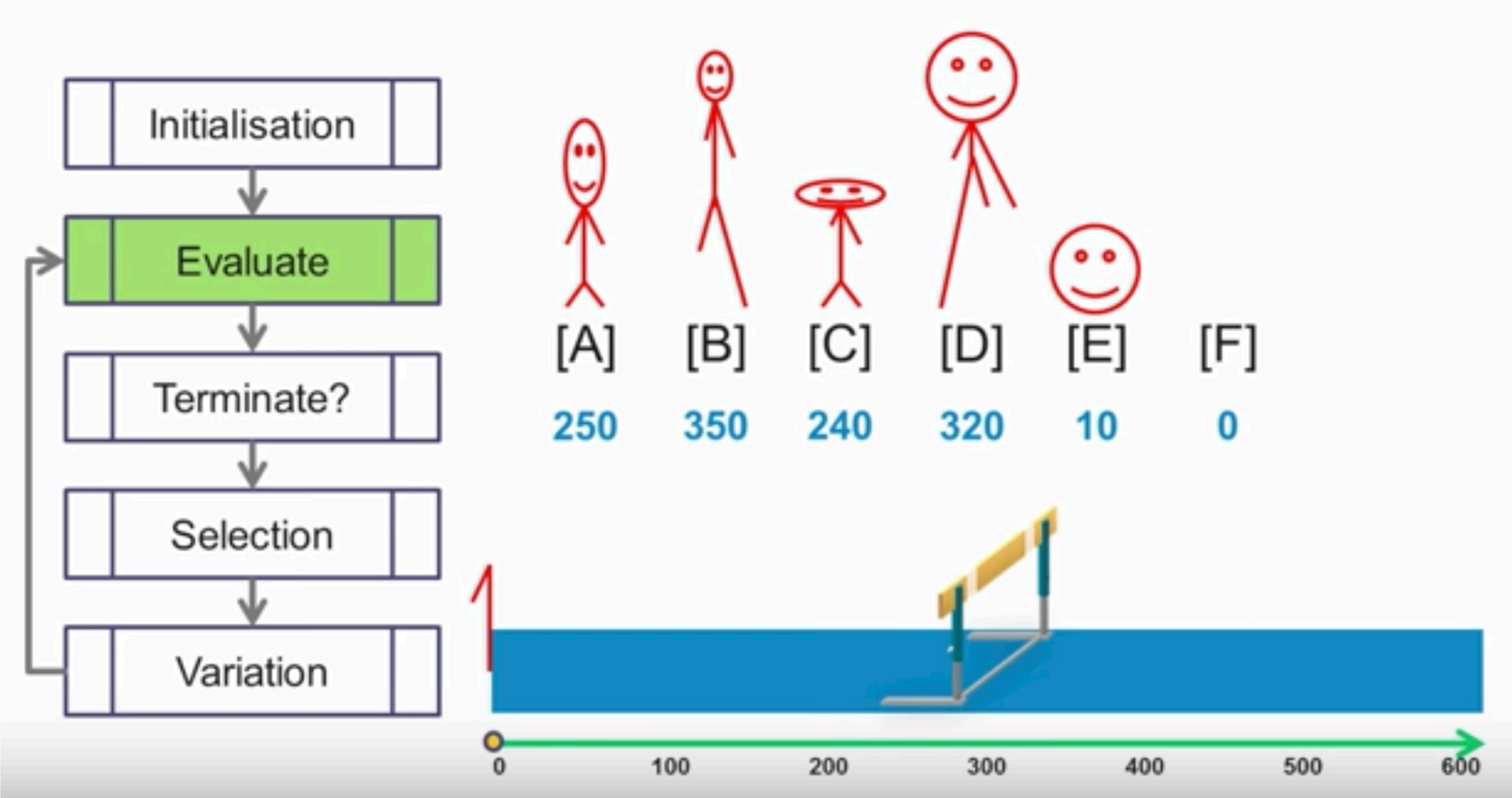
Source: Shahin Rostami (Bournemouth University)

Evolutionary Algorithms: how does it work?



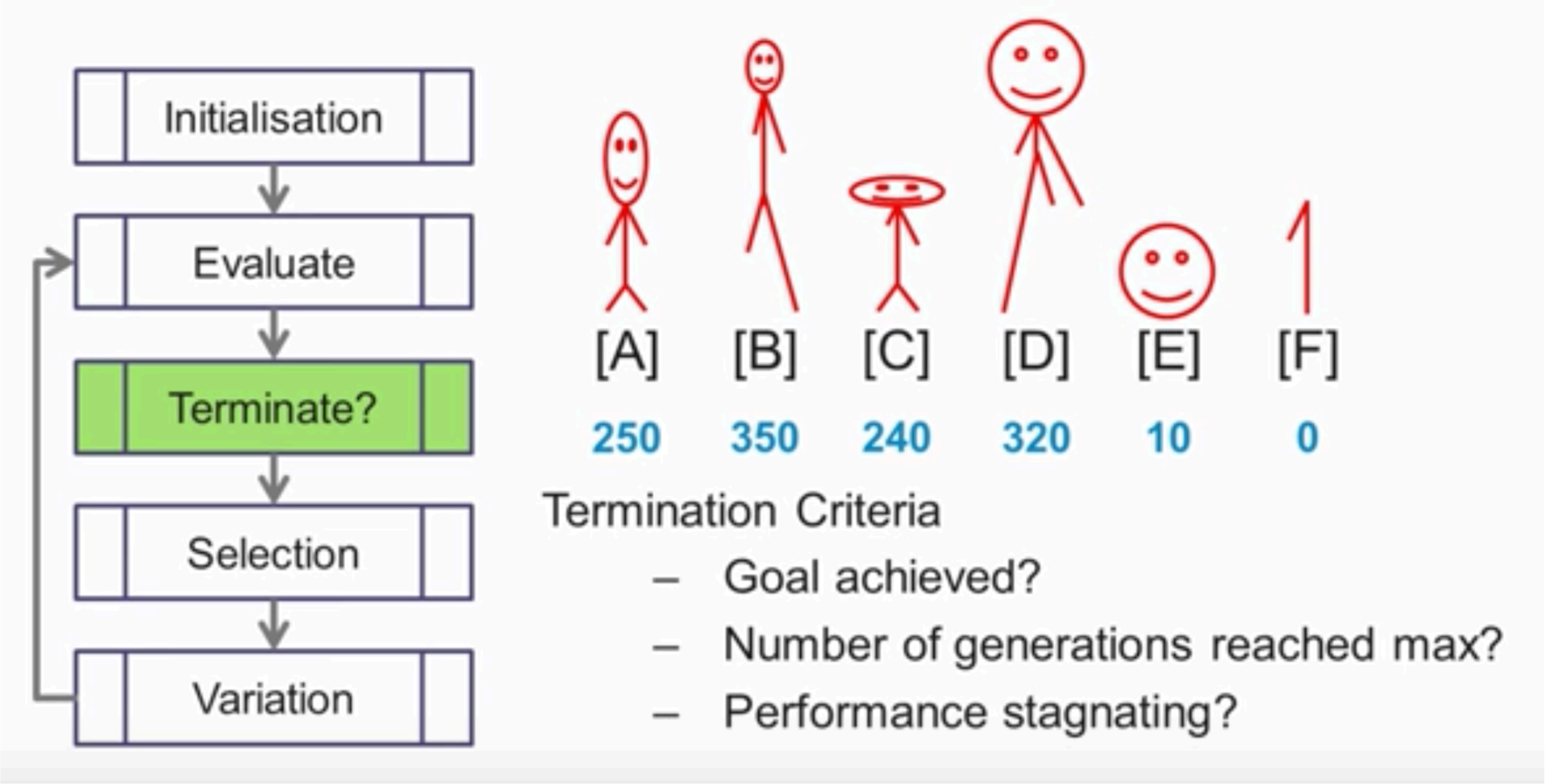
Source: Shahin Rostami (Bournemouth University)

Evolutionary Algorithms: how does it work?



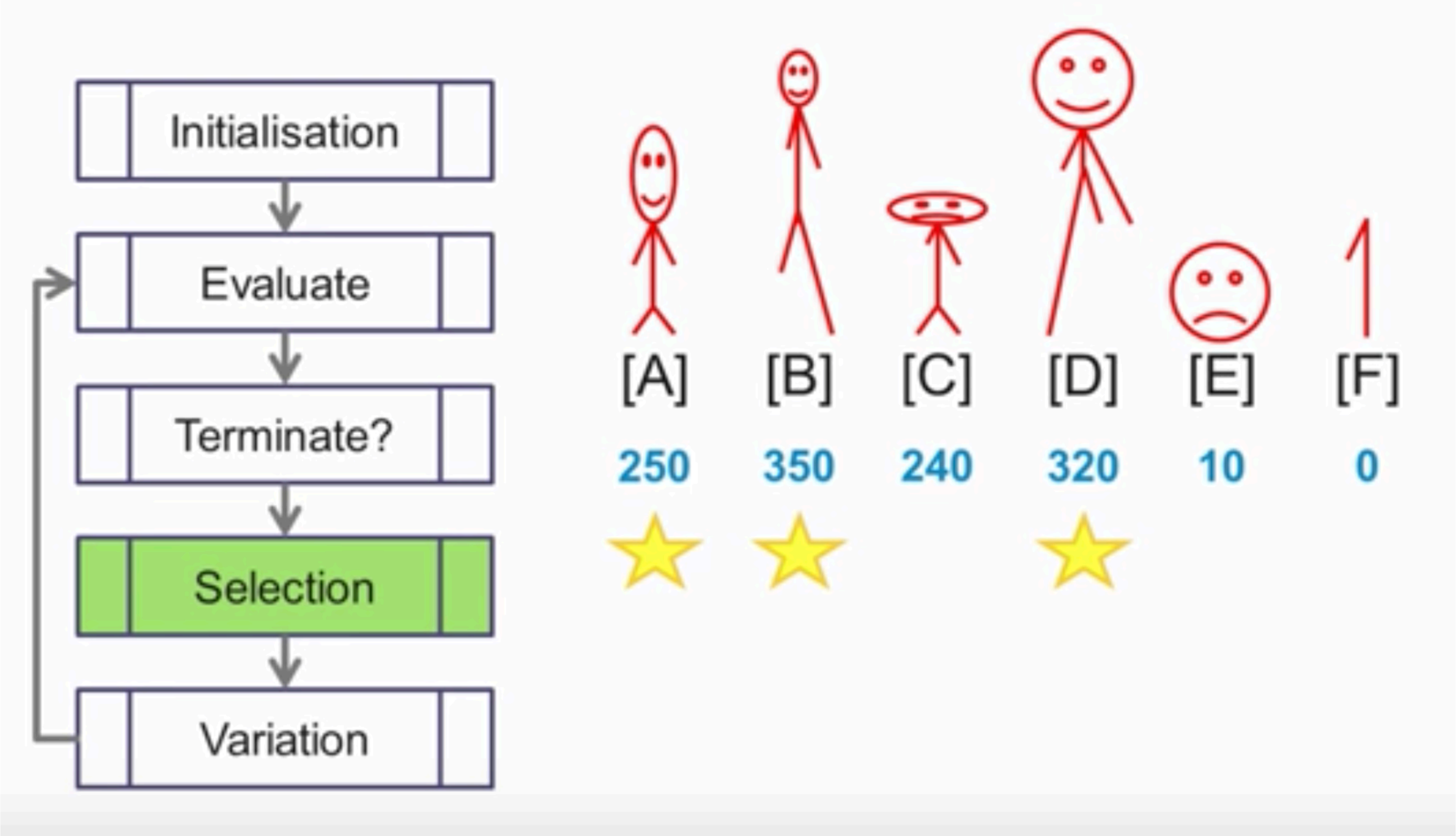
Source: Shahin Rostami (Bournemouth University)

Evolutionary Algorithms: how does it work?



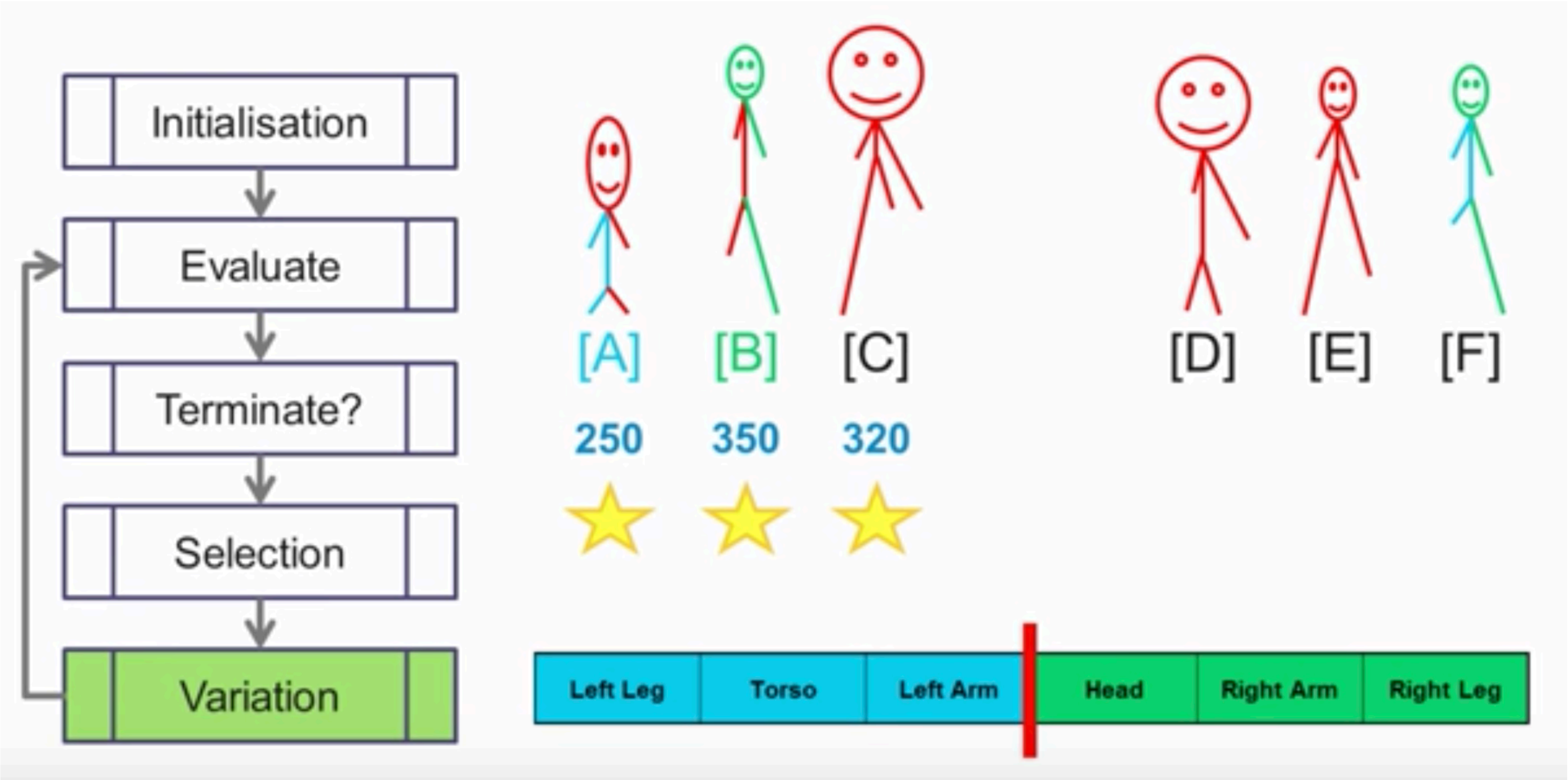
Source: Shahin Rostami (Bournemouth University)

Evolutionary Algorithms: how does it work?



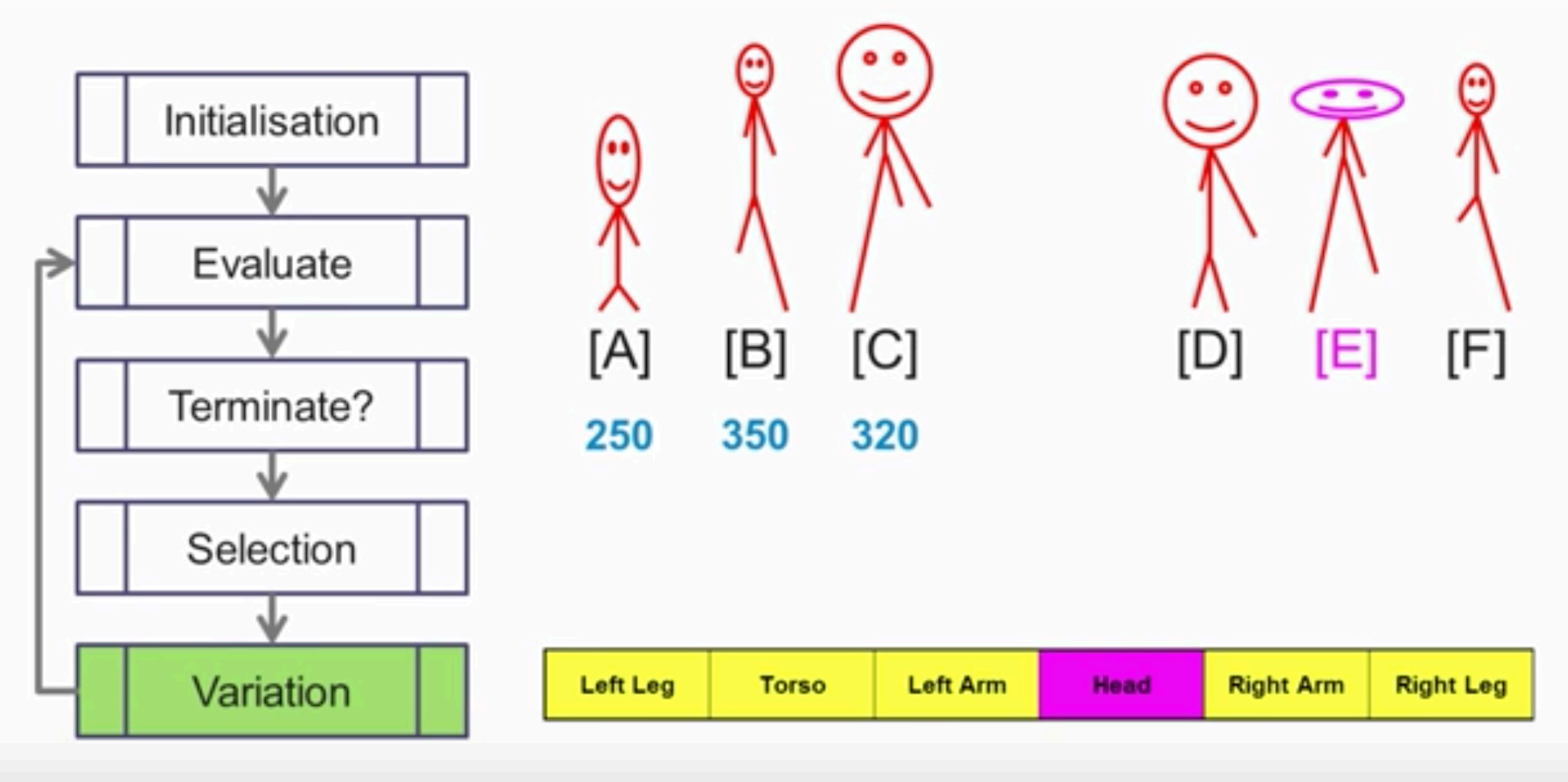
Source: Shahin Rostami (Bournemouth University)

Evolutionary Algorithms: how does it work?



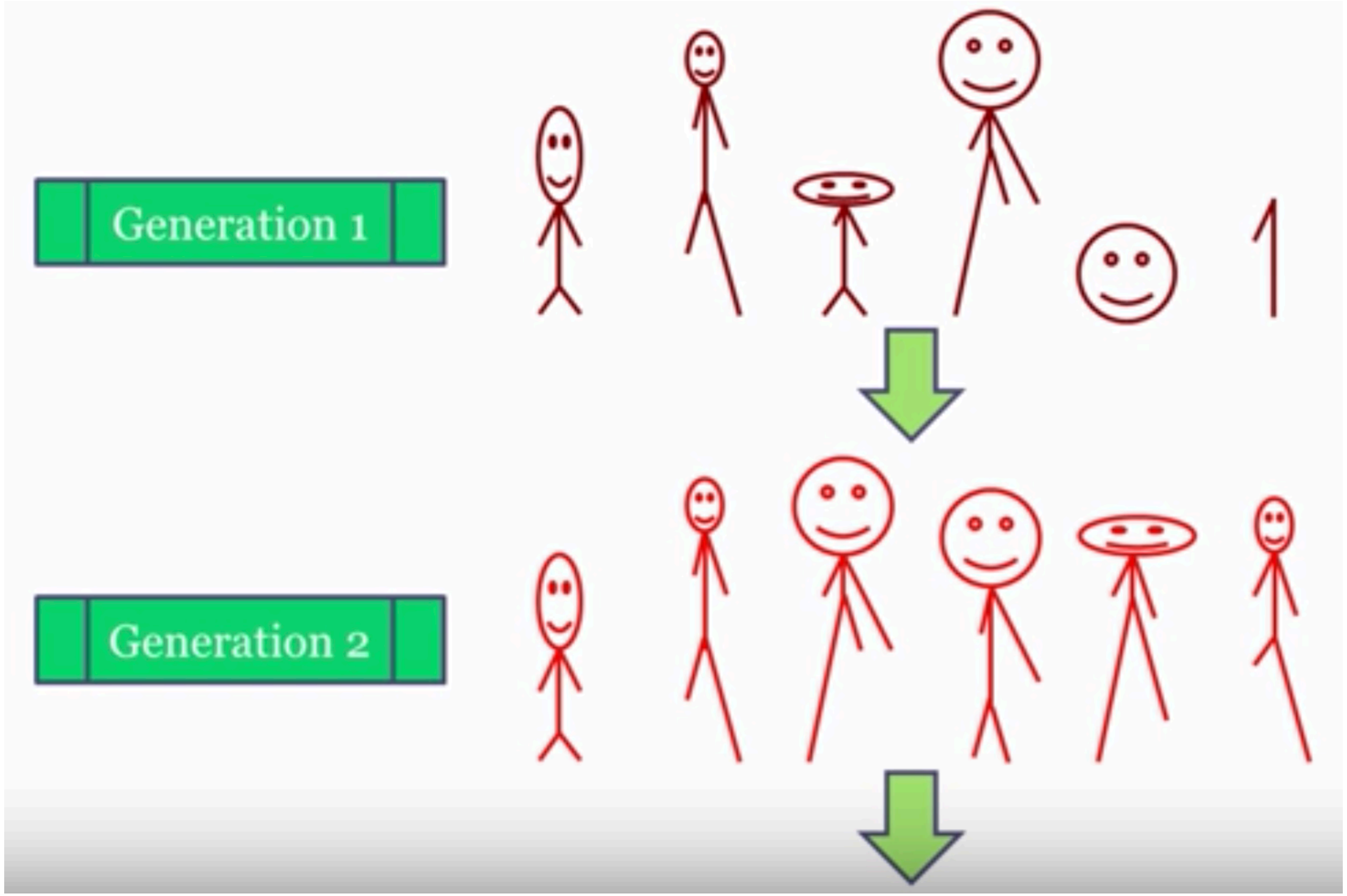
Source: Shahin Rostami (Bournemouth University)

Evolutionary Algorithms: how does it work?



Source: Shahin Rostami (Bournemouth University)

Evolutionary Algorithms: how does it work?



Source: Shahin Rostami (Bournemouth University)

Neuroevolution

Neuroevolution

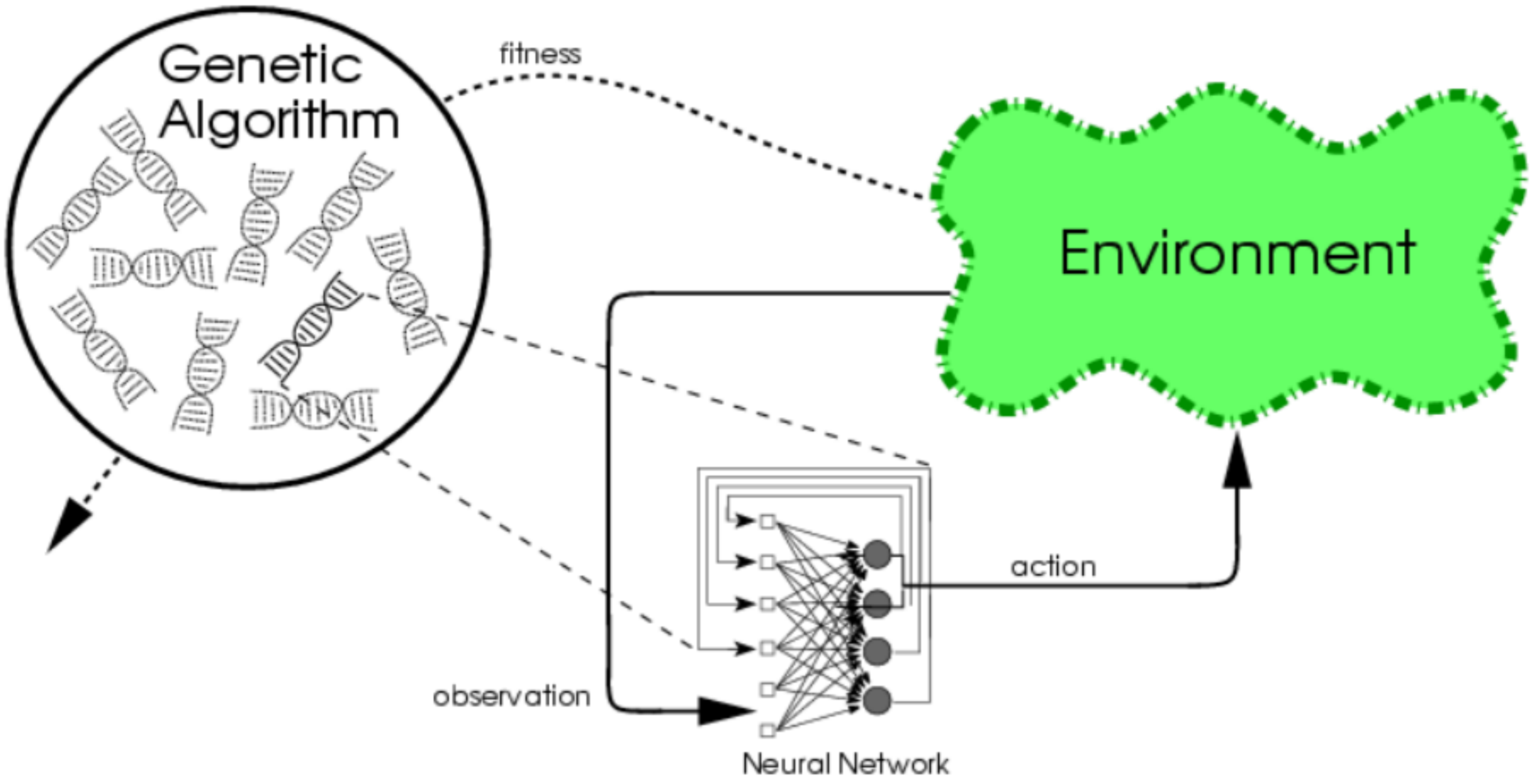
Joel Lehman and Risto Miikkulainen (2013), Scholarpedia, 8(6):30977.

- **Dr. Joel Lehman**, The University of Texas at Austin, Austin, TX, USA
- **Prof. Risto Miikkulainen**, The University of Texas at Austin, Austin, TX, USA

- **Neuroevolution** is a machine learning technique that applies evolutionary algorithms to construct artificial neural networks, taking inspiration from the evolution of biological nervous systems in nature.
- Neuroevolution is an effective approach to solving *reinforcement learning* problems, and is most commonly applied in *evolutionary robotics* and artificial life.

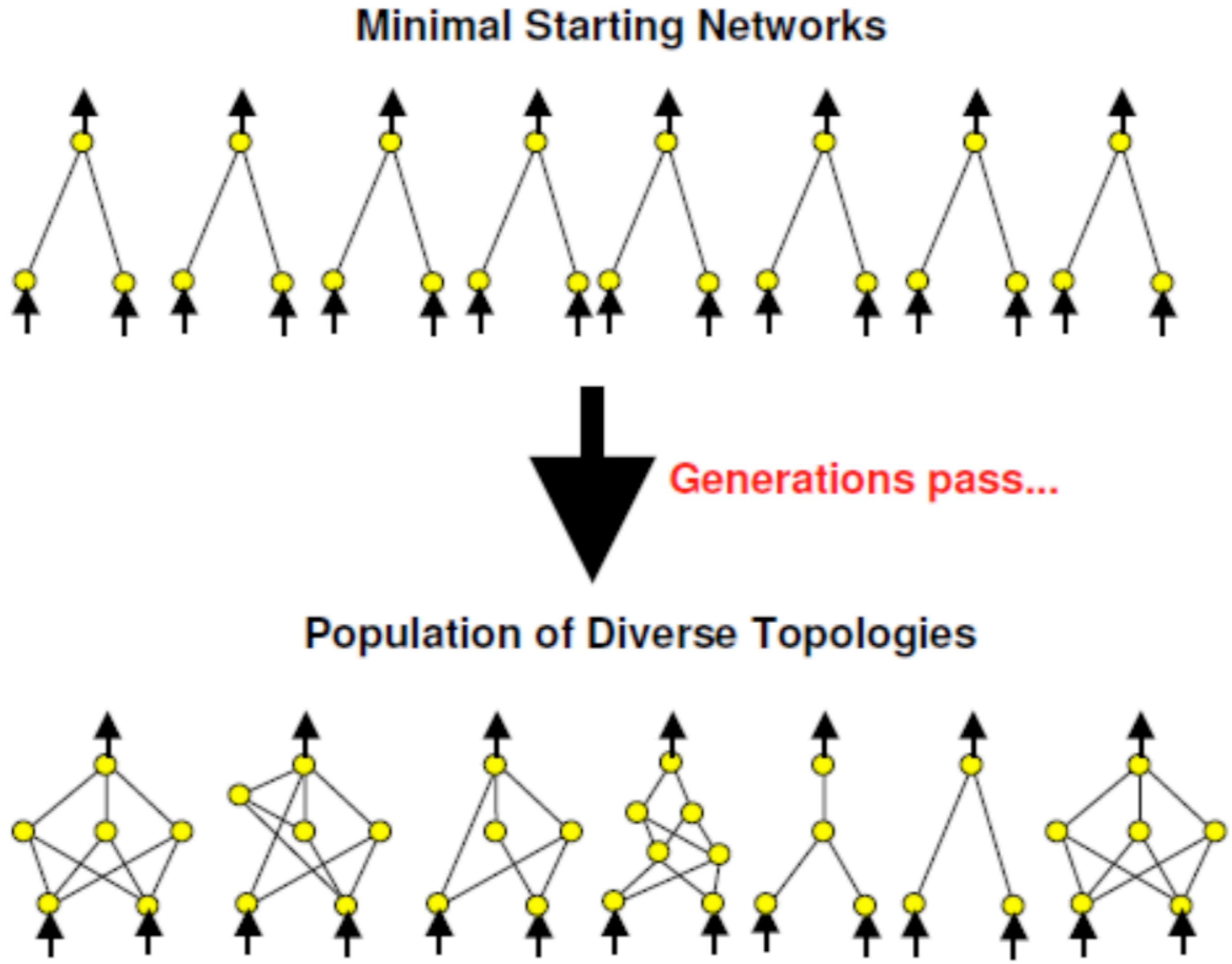
Source: http://www.scholarpedia.org/article/Neuroevolution

Conventional neuroevolution (CNE)



Source: http://www.scholarpedia.org/article/Neuroevolution

Neuroevolution: evolving topologies

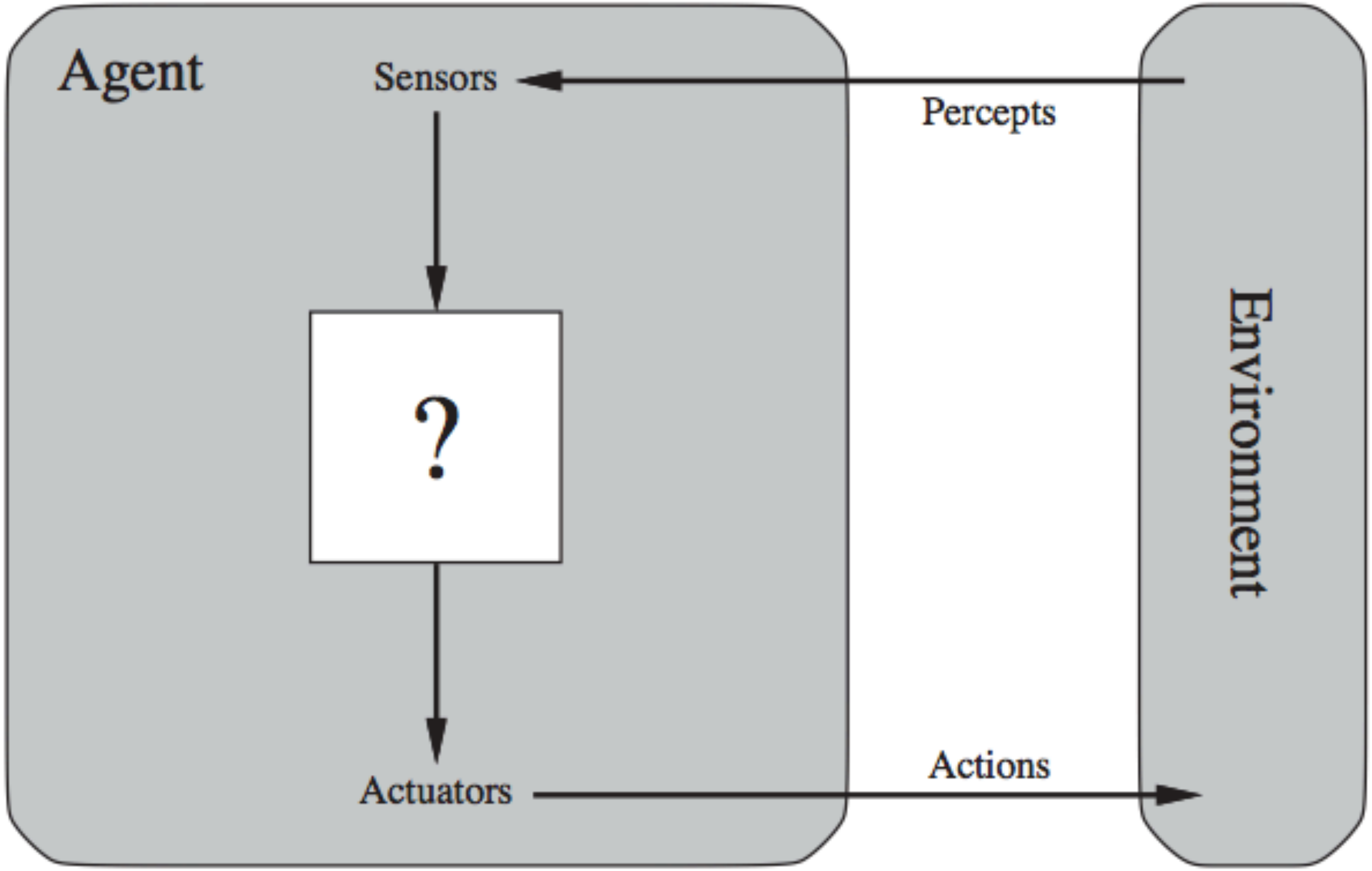


Neuroevolution: representative algorithms

- Baseline: fixed topology ANNs (1990s)
  - Topology and Weight Evolving ANNs (TWEANNs)
- NeuroEvolution of Augmenting Topologies (NEAT) [Stanley and Miikkulainen, 2002]
  - Variable (expanding) topologies
  - Ability to keep considering mutations with future potential
- HyperNEAT [Stanley, D'Ambrosio, and Gauci, 2009]
  - Compositional pattern-producing networks (CPPNs): indirect encoding (number of genes << number of connections)
- Novelty Search [Lehman and Stanley, 2011]
  - Parents should not always be selected based on their objective performance (fitness), but rather based on their novelty.

Reinforcement Learning (RL)

Reinforcement Learning (RL): agents



- A *rational agent* is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome.

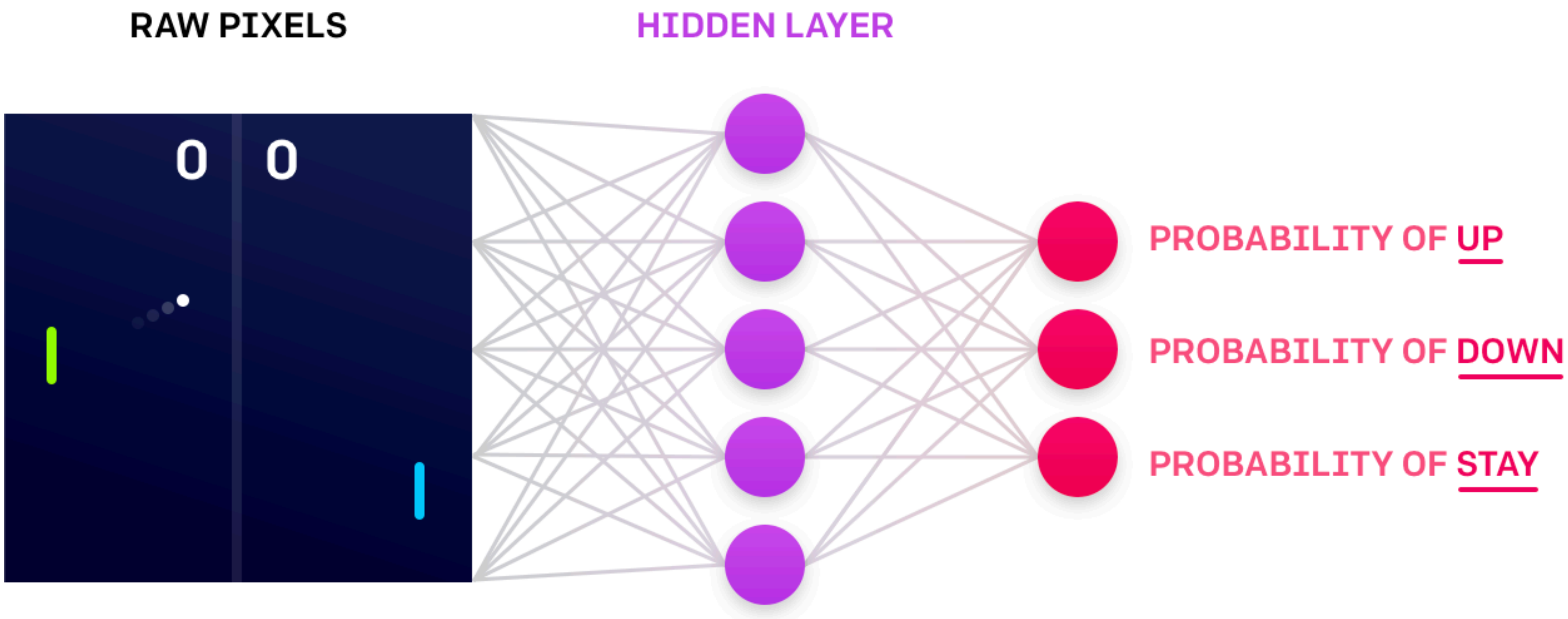
Source: Russell and Norvig (2010)

Reinforcement Learning (RL)

- Suppose we are given some *environment* (e.g. a game) that we'd like to train an *agent* on.
- To describe the *behavior* of the agent, we define a ***policy function*** (the brain of the agent), which computes how the agent should act in any given situation.
- In practice, the policy is usually an ANN that takes the current state of the game as an input and calculates the probability of taking any of the allowed actions.
- A typical policy function might have about 1 M parameters, so the task is to find the precise setting of these parameters such that the policy plays well (i.e. wins a lot of games).

Source: https://blog.openai.com/evolution-strategies/

Reinforcement Learning (RL)

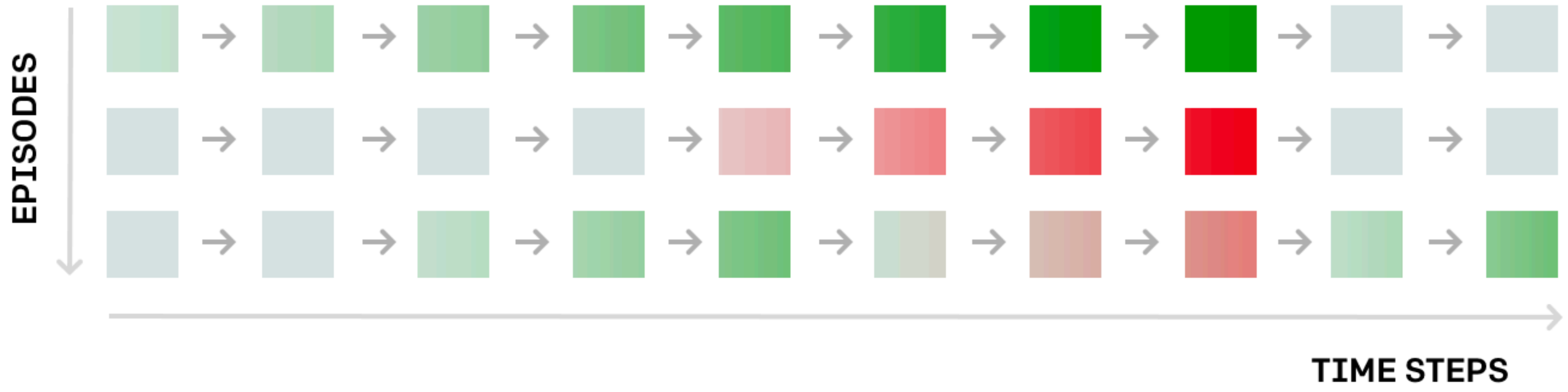


In the game of Pong, the ***policy*** *could take the pixels of the screen and compute the probability of moving the player's paddle (in green, on right) Up, Down, or neither.*

Source: https://blog.openai.com/evolution-strategies/

Reinforcement Learning (RL): training process

- Starting from a random initialization, let the agent interact with the environment for a while and collect episodes of interaction (e.g. each episode is one game of Pong).
  - We thus obtain a complete recording of what happened: what sequence of states we encountered, what actions we took in each state, and what the reward was at each step.



Source: https://blog.openai.com/evolution-strategies/

Reinforcement Learning (RL): learning

- Use backpropagation to compute a small update on the network's parameters that would make the green actions more likely in those states in the future, and the red actions less likely in those states in the future.
  - We expect that the updated policy works a bit better as a result.
- Note: the policies used in RL are typically stochastic, in that they only compute probabilities of taking any action.
  - This way, during the course of training, the agent may find itself in a particular state many times, and at different times it will take different actions due to the sampling.
- We then iterate the process: collect another batch of episodes, do another update, etc.

Source: https://blog.openai.com/evolution-strategies/

Reinforcement Learning (RL): benchmarks

- OpenAI Gym
  - Classic control theory problems
  - Atari 2600 games
  - 2D and 3D robots
    - MuJoCo (**Mu**lti-**Jo**int dynamics with **Co**ntact) http://www.mujoco.org/
  - … and more

Gym

Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.

https://gym.openai.com/

Reinforcement Learning (RL): baselines

- OpenAI Baselines: a set of high-quality implementations of the most representative RL algorithms in the literature.

  - "These algorithms will make it easier for the research community to replicate, refine, and identify new ideas, and will create good baselines to build research on top of."

OpenAI

Sidebar^2

- **OpenAI Under Fire**
  - An icon of idealism in AI stands accused of letting its ambition eclipse its principles.

- Read more at: https://www.deeplearning.ai/thebatch/ (Feb 26, 2020)

Deep Neuroevolution: selected recent examples (suggested reading)

Representative recent results

- Open AI “Evolution Strategies as a Scalable Alternative to Reinforcement Learning” [Salimans et al., 2017]
    - “We’ve discovered that evolution strategies (ES), an optimization technique that’s been known for decades, rivals the performance of standard reinforcement learning (RL) techniques on modern RL benchmarks (e.g. Atari/MuJoCo), while overcoming many of RL’s inconveniences.”
    - “In particular, ES is simpler to implement (there is no need for backpropagation), it is easier to scale in a distributed setting, it does not suffer in settings with sparse rewards, and has fewer hyperparameters.”

Representative recent results

- Uber AI Labs: suite of 5 papers (Dec 2017)
  1. Deep Neuroevolution: Genetic Algorithms are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning [Such et al., 2017]
  2. Safe Mutations for Deep and Recurrent Neural Networks through Output Gradients [Lehman et al., 2017]
  3. On the Relationship Between the OpenAI Evolution Strategy and Stochastic Gradient Descent [Zhang, Clune, and Stanley, 2017]
  4. ES Is More Than Just a Traditional Finite Difference Approximator [Lehman et al., 2017]
  5. Improving Exploration in Evolution Strategies for Deep Reinforcement Learning via a Population of Novelty-Seeking Agents [Conti et al., 2017]

Late addition

ACADEMIC PRESS

Artificial Intelligence in the Age of Neural Networks and Brain Computing

2019, Pages 293-312

Chapter 15 - Evolving Deep Neural Networks

Risto Miikkulainen [1,2], Jason Liang [1,2], Elliot Meyerson [1,2], Aditya Rawal [1,2], Daniel Fink [1], Olivier Francon [1], Bala Raju [1], Hormoz Shahrzad [1], Arshak Navruzyan [1], Nigel Duffy [1], Babak Hodjat [1]

⊞ Show more

https://doi.org/10.1016/B978-0-12-815480-9.00015-3

Resources

Hands on!

- Code for NEAT / HyperNEAT / Novelty Search: http://eplex.cs.ucf.edu/neat_software/
- Code for OpenAI's "Evolution Strategies as a Scalable Alternative to Reinforcement Learning" [Salimans et al., 2017]: https://github.com/openai/evolution-strategies-starter
- Code from Uber papers:
  - Deep Neuroevolution: Genetic Algorithms are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning [Such et al., 2017] and
  - Improving Exploration in Evolution Strategies for Deep Reinforcement Learning via a Population of Novelty-Seeking Agents [Conti et al., 2017]: https://github.com/uber-research/deep-neuroevolution
  - Safe Mutations for Deep and Recurrent Neural Networks through Output Gradients [Lehman et al., 2017]: https://github.com/uber-research/safemutations

Final words

- “Neuroevolution is the only branch of AI with an actual proof of concept: brains *did* evolve, so we know that's one way to produce intelligence.”

-- Kenneth O. Stanley (UCF and Uber AI Labs)

"We've always said that human-level intelligence is 20 years off.

Eventually we'll be right."

-- Patrick H. Winston, Professor of Artificial Intelligence and Computer Science at MIT