Simulierte Evolution: Hands-On-Starter Kit

JFS 2022

Heiko Spindler
Freelancer

iX DEVELOPER

Winter 2018

€ 12,90

Österreich € 14,20
Schweiz CHF 25,80
Benelux € 14,80
Italien € 16,80

www.ix.de

Machine Learning

Verstehen, verwenden, verifizieren

Data Science – Status quo

Neuronale Netze in Theorie und Praxis

Machine Learning as a Service: Cloud-Angebote

Herausforderungen bei Bild- und Texterkennung

Blick in die Blackbox

TensorFlow, Keras & Co.

ML-Frameworks und -Bibliotheken

Programmiersprachen: Python, Scala, C++

Hardware pusht ML

GPUs und CPUs für Machine Learning

Neuromorphe Chips für neuronale Netze

Mensch trifft KI

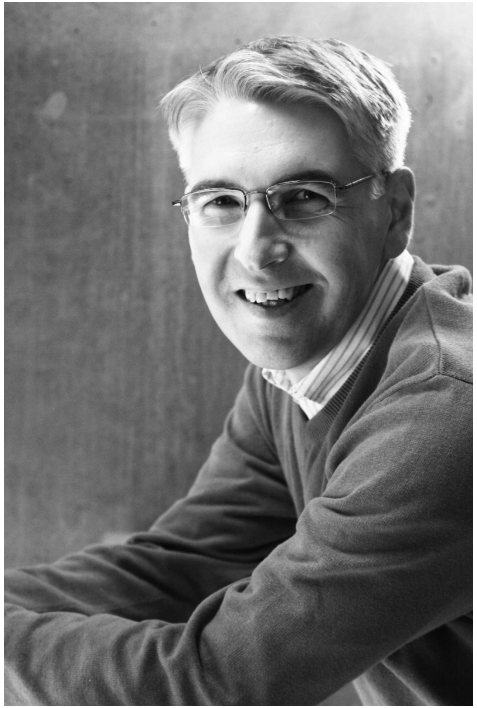
Rechtsfragen bei Künstlicher Intelligenz

Verantwortungsvoller Einsatz und Ethik

Heiko Spindler

Freiberuflicher IT-Berater,
Software-Entwickler
und Coach

Mail: hs@heikospindler.de



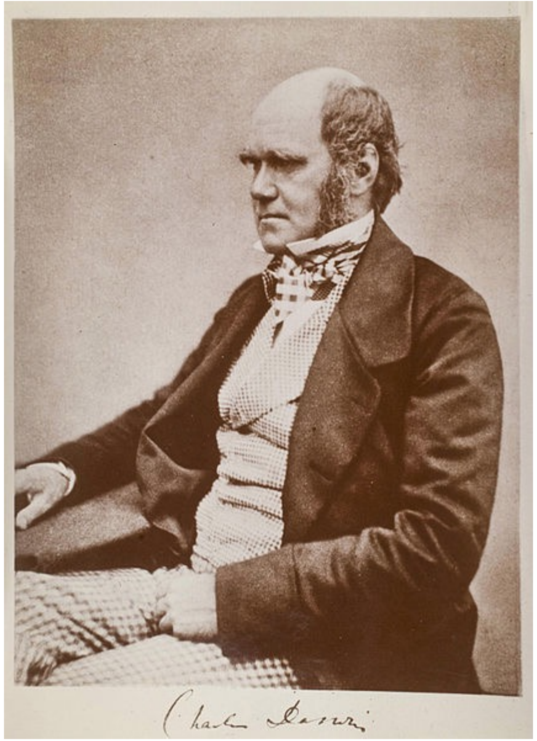
Evolution

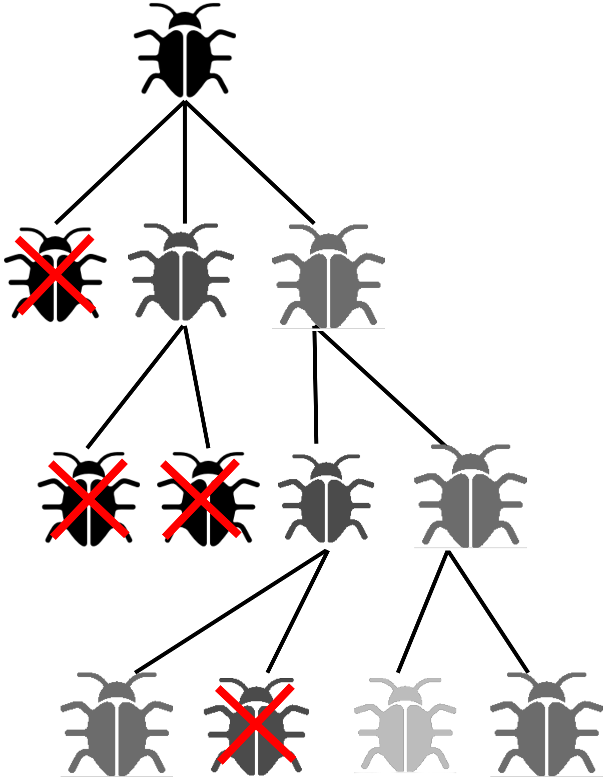
... is the process by which traits that enhance survival and reproduction become more common in successive generations:

Variation exists within populations of organisms: morphology, physiology, and behavior

Different traits confer different rates of survival and reproduction (differential fitness)

Traits can be passed from generation to generation (heritability of fitness).





Mutation creates variation

Not every offspring survives

Favorable mutations are more likely to survive and reproduce

Evolution
- not only in nature!

Do you know other areas where Evolution takes place?

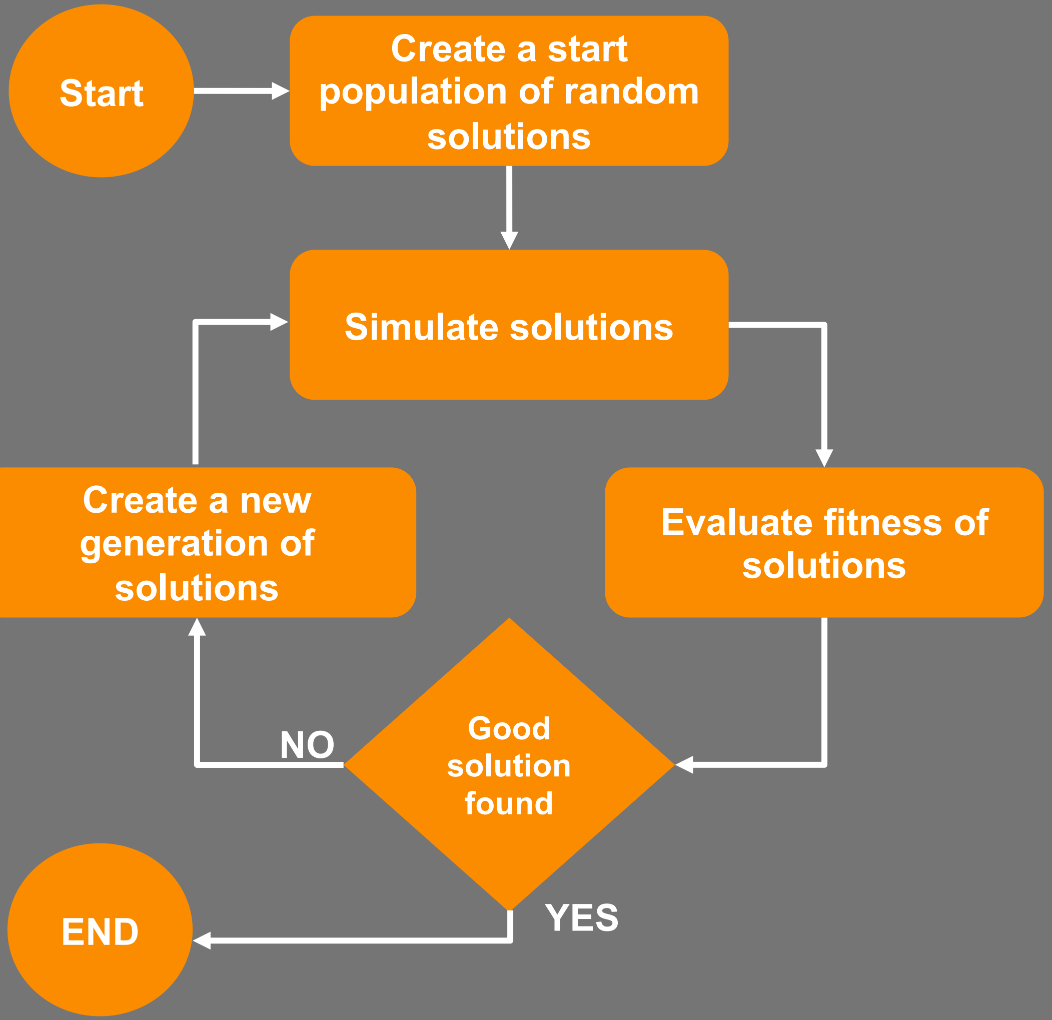
NOKIA
Connecting People

Know our past. Create the future...

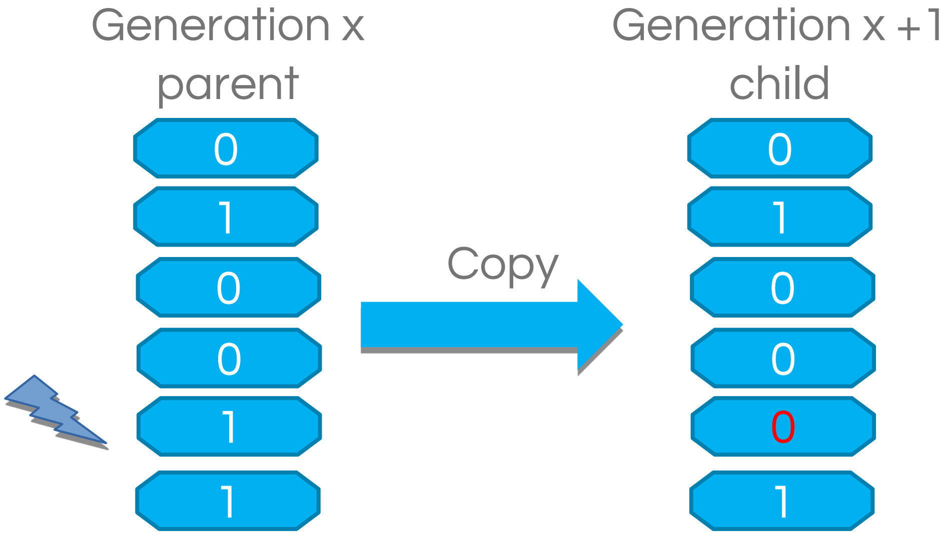
What do we need to simulate Evolution?

- Data structure for genes
- Implementation of genetic operations
- Implementation of the process
- Simulation of solutions
- Fitness function

Process of Evolution

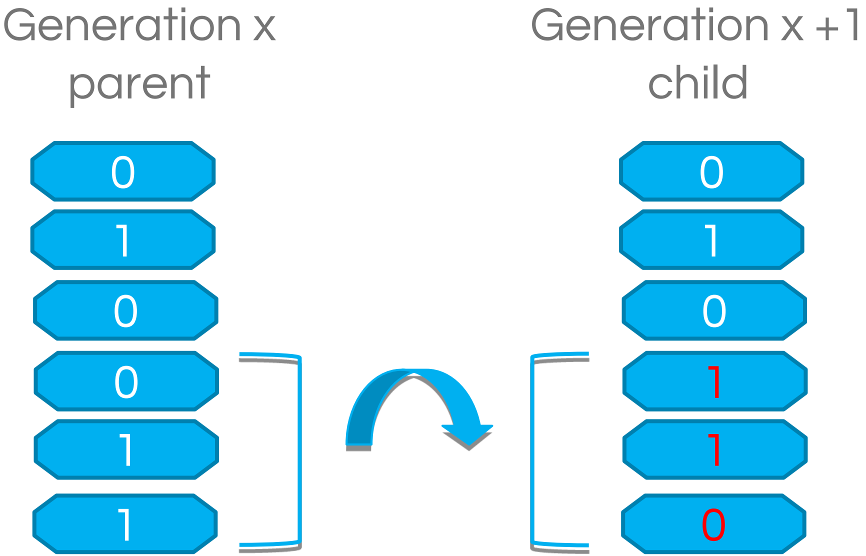

Start
Create a start population of random solutions
Simulate solutions
Evaluate fitness of solutions
Good solution found
NO
Create a new generation of solutions
YES
END

Mutation



Changes a value at a random position

Inversion



Generation x parent	Generation x +1 child
0	0
1	1
0	0
0	1
1	1
1	0

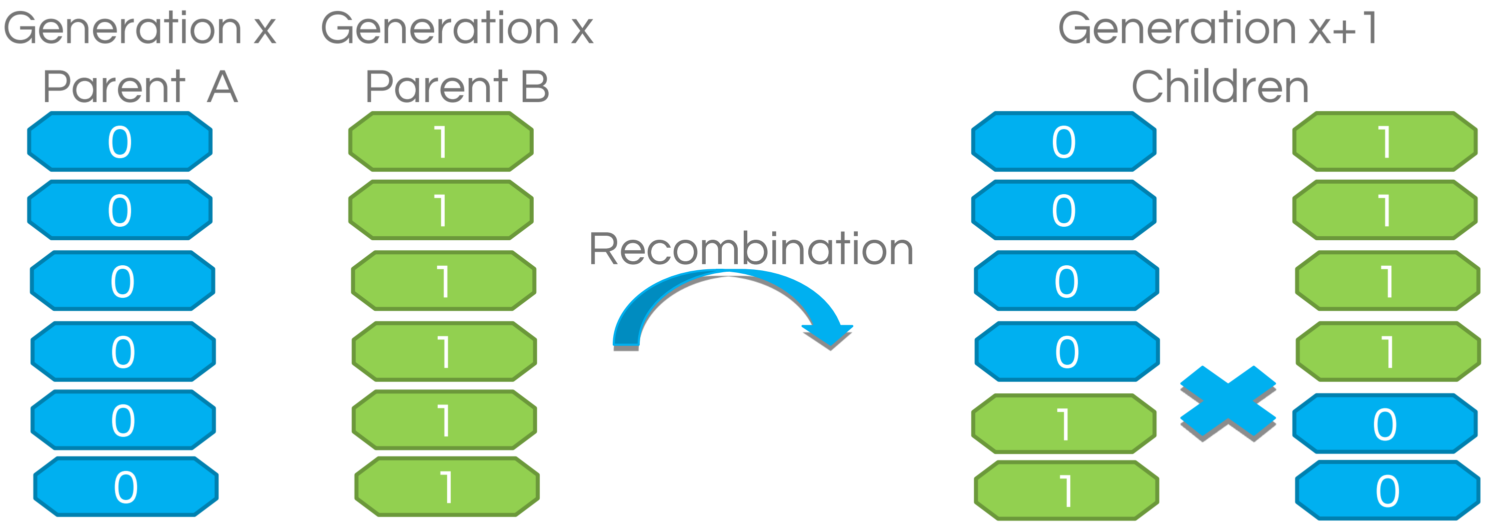
The sequence changes.

Recombination

Two solution mate

Fast spreading of good traits

Recombination / Cross-Over



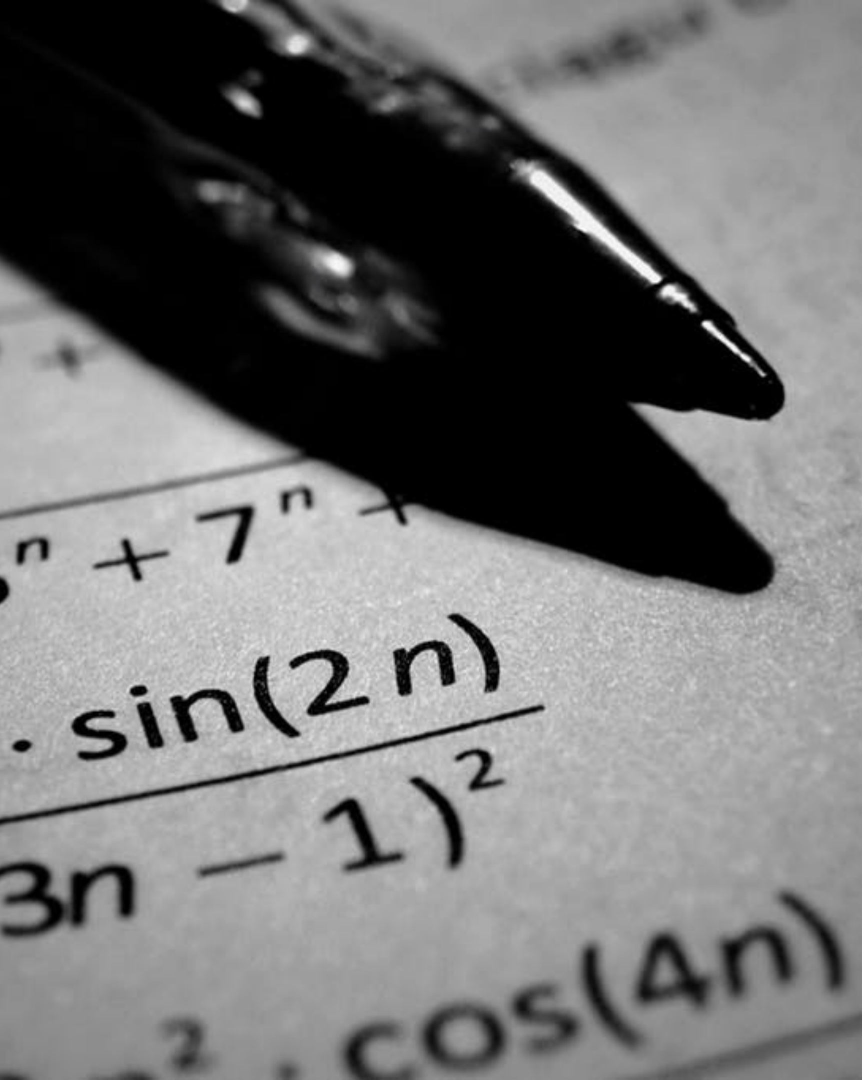
Generation x	Generation x		Generation x+1	
Parent A	Parent B		Children	
0	1		0	1
0	1		0	1
0	1	Recombination	0	1
0	1		0	1
0	1		1	0
0	1		1	0

To Dos

- Define an encoding / decoding for the genome
- Define a Fitness-Function
- Define good parameters for the evolution process:
    - Population size
    - Mutation rate
    - Cross over rate

Frameworks can help

- Create initial populations (random)
- Run the process
    - Calculate the fitness
    - Sort solutions by fitness
    - Apply genetic operators
    - Distribute the work



Define a Fitness Function

- Simple to calculate
- Maps a solution to a number (Higher number: better solution)

Frameworks

JGAP: http://jgap.sourceforge.net

~~Apache Commons Math: http://commons.apache.org~~

Jenetics: http://jenetics.io

Features

- Java, Open Source
- Data structures
- Genetic operations
- Interfaces for fitness functions
- Process of simulated evolution

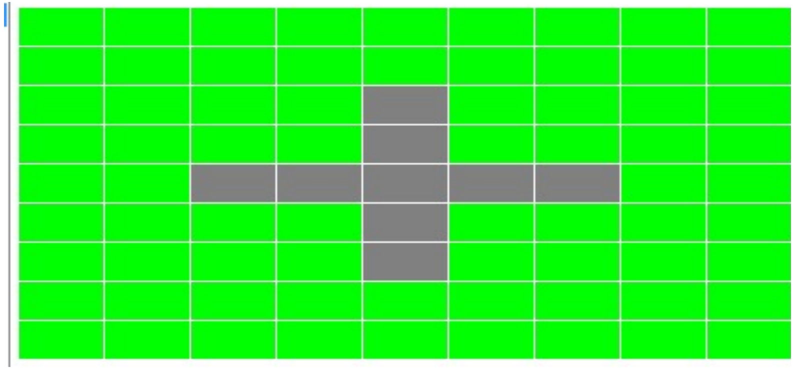
Sample Coding

The challenge

A Robot should visit and mark all cells of a given board (9 x 9 cells).

Some cells are blocked (gray).

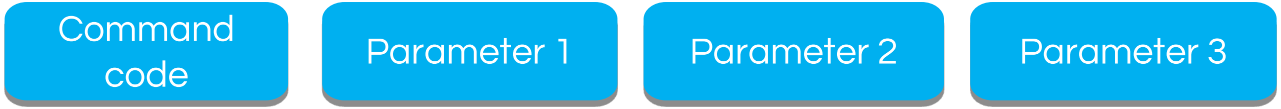
The memory holds up to 30 commands.



Commands

Command	Description
`Move` <left, right, up, down>	Moves the robot
`Set`	Marks the cell where the robot is located
`Nop`	„No Operation“
`Goto` <Command No.>	Continues the execution of the program at another position
`If` <left, right, up, down> = <free, marked, border> `Goto` <Command No.>	Conditional jump depends on the state of the current cell

Mapping of Commands to Genes



Command code

Parameter 1

Parameter 2

Parameter 3

Commands are stored in 4 integer values

Not used parameters are ignored

Fitness Function

Priority 1: „Get the job done“:

- Fitness is given by the percentage of marked cells (0.0 to 100.0).

…

Fitness Function

Priority 1: „Get the job done“:

- Fitness is given by the percentage of marked cells (0.0 to 100.0).

Priority 2: „Be efficient“:

- **Every command execution consumes energy**
- **The robot gets energy for new marked cell**
- **The energy left after marking all cells defines the fitness.**

Main parameters to drive evolution

Population Size

- 50 up to 1000 Individuals.
- In a large population, a slight improvement may not prevail and is lost in the crowd.

Mutation Rate

- High values result in much destruction.
- Too low values mean too little creativity and new ideas.

Crossing Over Rate

- High Rate: Process converges quickly but fixes (too) early on specific solutions.
- Niedrige Rate: Gute Individuen (insbesondere nur leichte Verbesserungen) setzen sich evtl. nur langsam oder gar nicht durch.

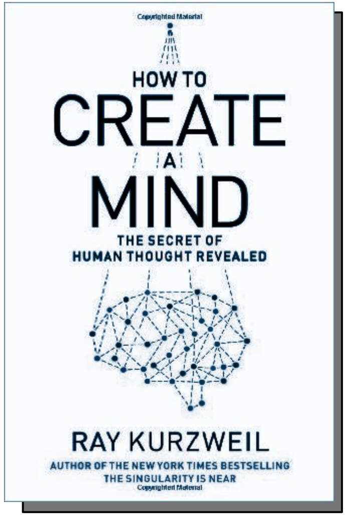
Ideas for Improvement

- Adaptive mutation rate:
  Long periods of stagnation lead to an increase in the mutation rate.
- Elitism:
  The best solution(s) will always survive (copied to next generation).
- Island Model:
  Separated populations breed different solutions and share the best individuals from time to time.

Simulated Evolution in Action

Ray Kurzweil, 2016

“How to Create a Mind: The Secret of Human Thought Revealed”



Copyrighted Material

HOW TO CREATE A MIND

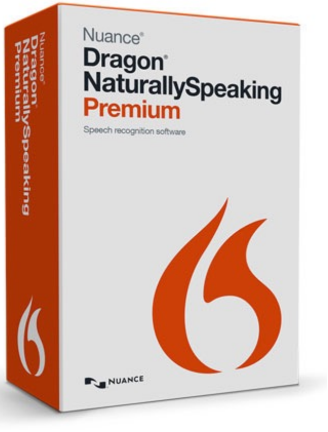
THE SECRET OF HUMAN THOUGHT REVEALED

RAY KURZWEIL

AUTHOR OF THE NEW YORK TIMES BESTSELLING THE SINGULARITY IS NEAR

Copyrighted Material

NUANCE®


Nuance®
Dragon®
NaturallySpeaking
Premium
Speech recognition software
NUANCE
Nuance® Dragon® NaturallySpeaking Premium

Q & A

Heiko Spindler

Freelacer
Mail: hs@heikospindler.de