

Machine Learning Using Karoo GP

ASPIRE WORKSHOP

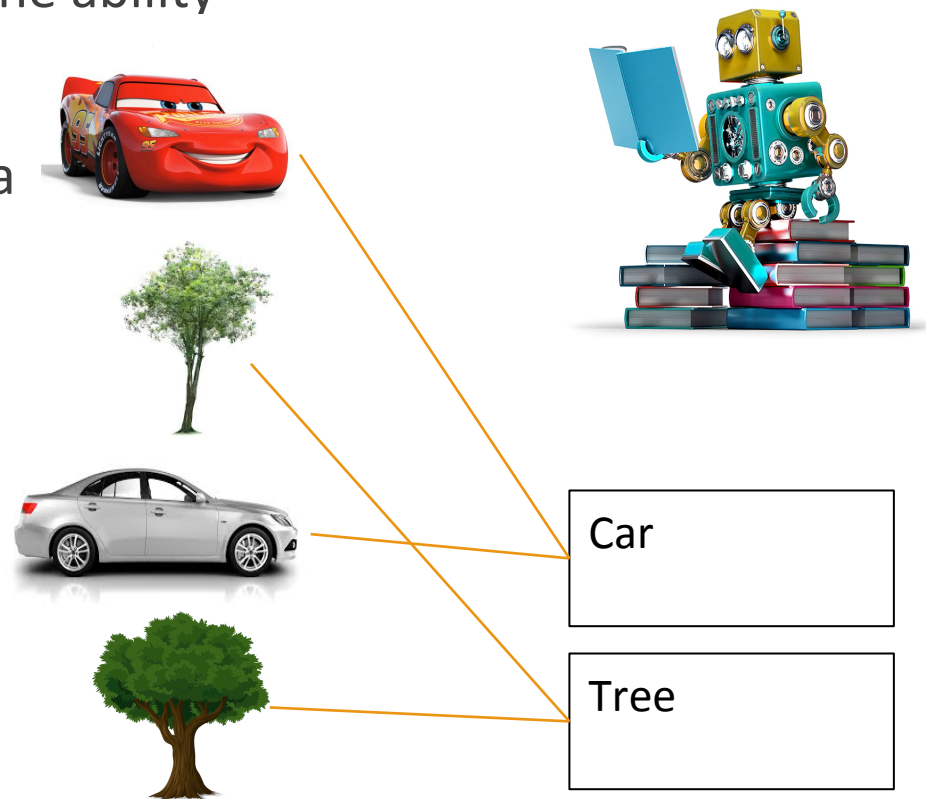
SUMMER 2017

THE OHIO STATE UNIVERSITY



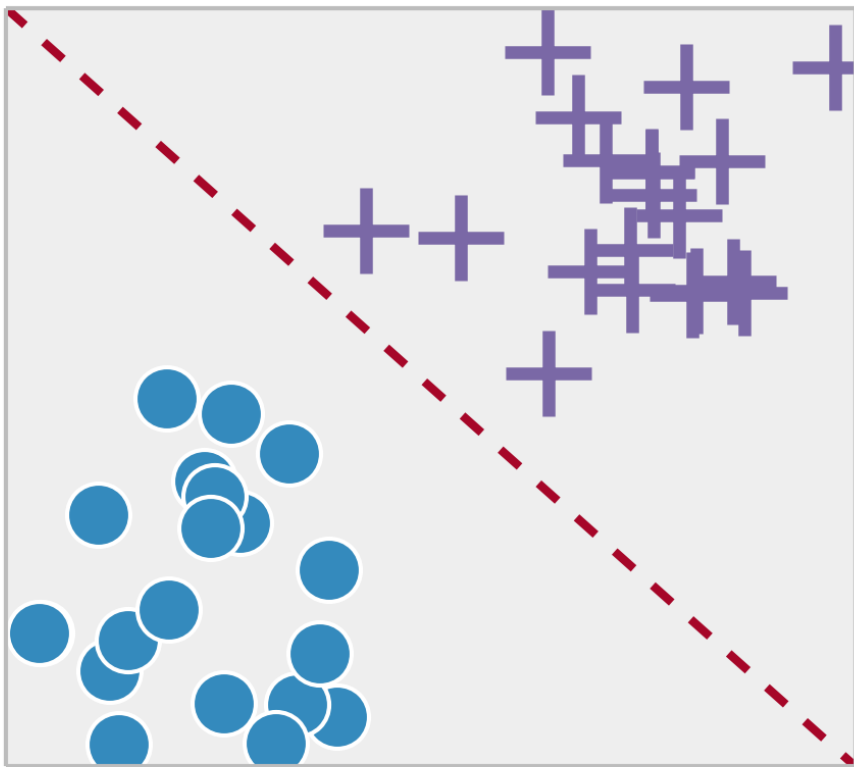
What is machine learning?

- Machine learning (ML) gives computers the ability to learn without explicit programming
- The goal is to find a structure that maps a given input to a desired output
- Examples of problems ML could solve:
 - Netflix movie recommendations
 - Categorizing images
 - Online advertising
 - Fitting data to a function

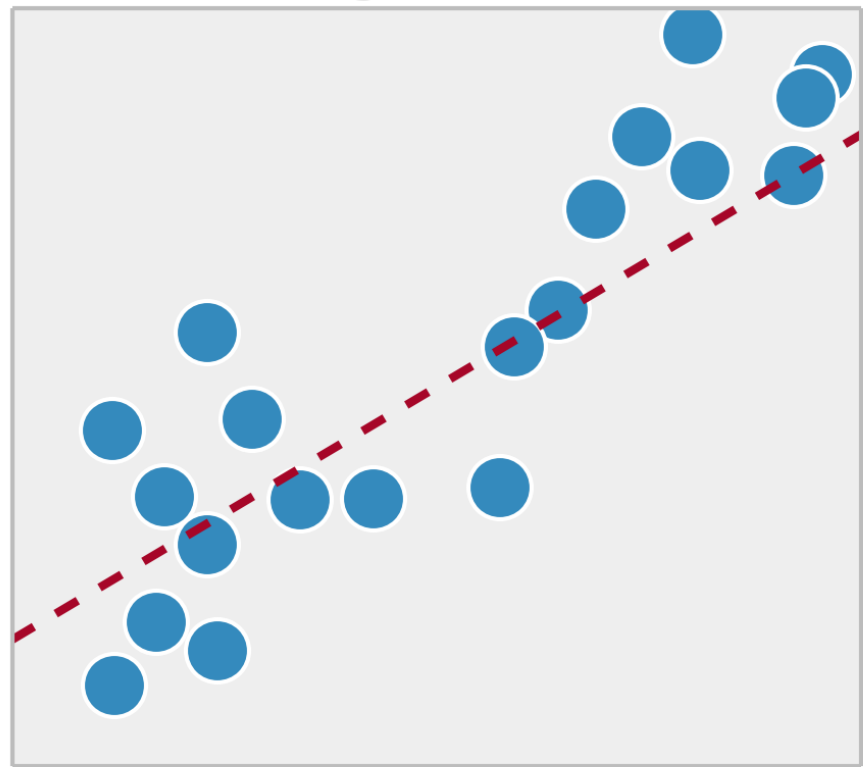


Regression vs. Classification

Classification



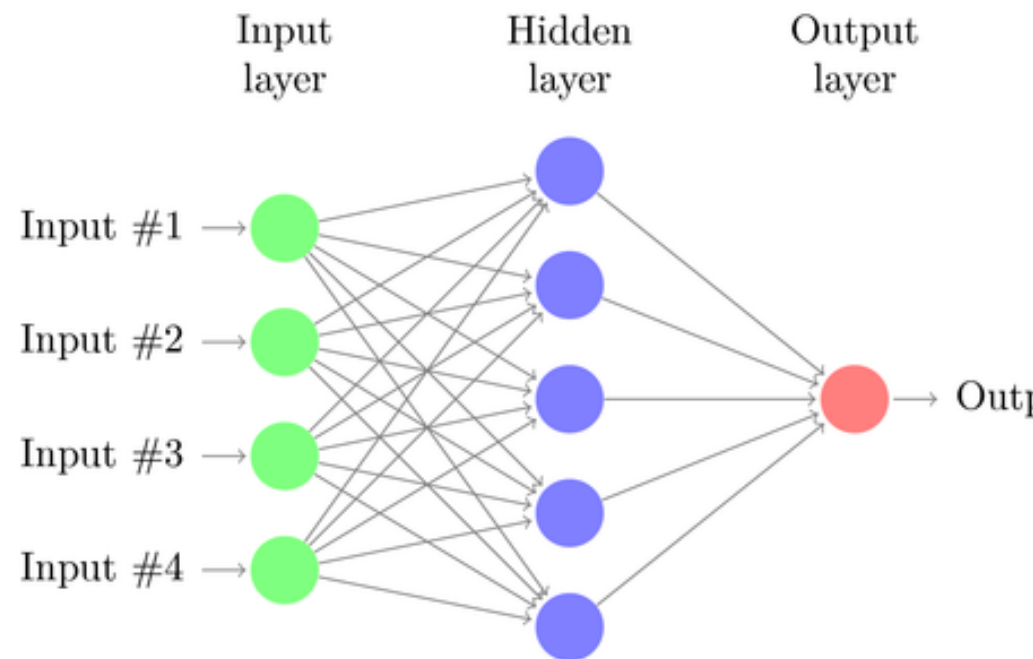
Regression



Some approaches to machine learning

Artificial Neural Networks:

- They work like neurons in our brains!
- The hidden layer uses a series of operations to take a set of inputs and turn it into the desired output, similar to how a brain processes data



Genetic Programming (GP)

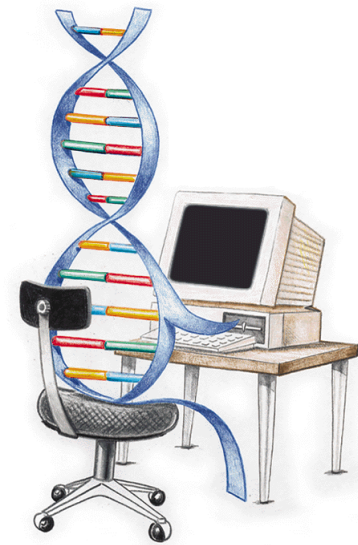
What would happen if we applied evolution to a Regression Problem?

Genetic programming approach:

- Create a group of random solutions

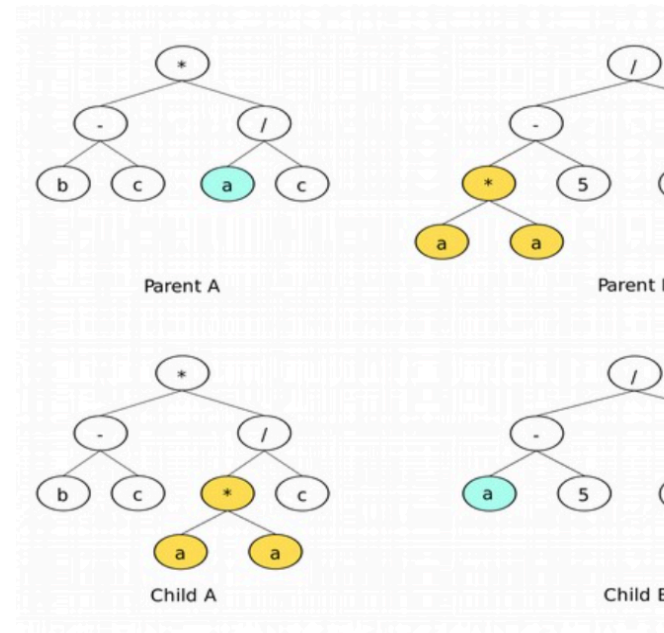
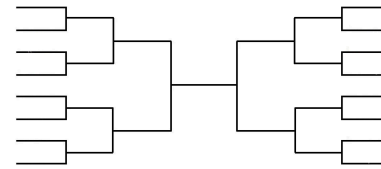
Ex: 1st solution = $2x^3 + e^y$, 2nd solution = $\cos(x)/x$, etc.

- Test and evaluate each solution
- Keep the best solutions and “kill” the rest
- Mutate the best solutions to produce even better ones



Genetic Programming (Cont.)

- Each solution is tested based on a fitness function
 - For Regression: Uses a least squares method
 - For Classification: Uses a relative absolute error
- The best solutions are chosen through a tournament selection
- The mutation process works like the following:
 - Reproduction: The function is copied to the next generation
 - Point Mutation: One thing is change ex: $2*t/(x+y)$ to $2*t/(x+t)$
 - Branch Mutation: A full branch is changed ex: $2*t/(x+y)$ to $2*t/(y^2*t)$
 - Crossover: 2 parents combine to produce offspring (see picture)



Karoo GP

- Kai Staats created a GP software called Karoo GP
- Karoo was Kai's master's thesis at the University of Cape Town
- Karoo is used to mitigate backgrounds for the LIGO gravitational wave observatory
- In addition to Genetic Programming Kai also enjoys rock climbing, making documentaries, creating radio telescopes and attending Space University



Karoo GP Cont.

Karoo GP is used to perform various tasks:

- Regression: Find a relationship between two or more variables
Ex: $s = 2x^2 + 5y + z^4$
- Classification: Assign variables to different categories
Ex: Distinguish between pictures of roses, sunflowers and a peonies
- Matching: Find a solution that matches the output with the inputs
Ex: $a+b+c = s$.

```
** ** ***** ** ** ***** ** ** ***** ** ** ***** ** **  
** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **   ** ** ** **   ** ** **   ** ** **   ** ** **  
** ** ***** ** ** ***** ** ** ***** ** ** ***** ** **  
** ** ** **   ** ** **   ** ** **   ** ** **   ** ** **  
** ** ***** ** ** ***** ** ** ***** ** ** ***** ** **  
** ** ** **   ** ** **   ** ** **   ** ** **   ** ** **
```

Genetic Programming in Python - by Kai Staats, version 0.9.1.6

We have constructed a population of 100 Trees for Generation 1

Evaluate the first generation of Trees ...

```
Tree 1 yields (sym): -4*a + 2*b + b/c**2 + c - 1  
Tree 2 yields (sym): -a/b - b*c - b - c - 2*c/b  
Tree 3 yields (sym): a*b**3/c + a + b - 1 + c**2/a  
Tree 4 yields (sym): -a*c + 2*a + b**2 + b/c + c**2 + 2*c - 1  
Tree 5 yields (sym): -a**2 + 2*c - c/a - c/(a*b**2)  
Tree 6 yields (sym): a**3*b/c + a + b + 2*c + 1/a  
Tree 7 yields (sym): a*b/c + a*c + 2*a + b + c**2 - c/a  
Tree 8 yields (sym): a*b + b**2*c + b + c**2 - c - b*c/a  
Tree 9 yields (sym): -b - c**2 - c - b/a  
Tree 10 yields (sym): -2*a - 2*b + c - 1 + c/a + c/a**2  
Tree 11 yields (sym): a**2*b + a*b*c + a + b + c + b/(a*c)  
Tree 12 yields (sym): -2*a - b*c + c - 1/(b*c**2)  
Tree 13 yields (sym): -b**2*c + b - b/c - c**2 + 2  
Tree 14 yields (sym): a*b + 3*a + 2*c + b**(-3) + b/a  
Tree 15 yields (sym): a*b + 4*b + 2*c + b/a + c/a  
Tree 16 yields (sym): -a*b + a*c**2 + b*c/a  
Tree 17 yields (sym): a*b/c + b*c + 2*b + c  
Tree 18 yields (sym): -a + a/c + b**2/c + 1  
Tree 19 yields (sym): -a*b - a + b**2 + b + b/c + c + b/(a*c)  
Tree 20 yields (sym): -a*b - a - a/c - a/b**2 + b*c + b + c  
Tree 21 yields (sym): a**2*b*c + a*b - c - b*c/a  
Tree 22 yields (sym): -a*c**2 - a - a/c + a/b + 3*b + c - 1  
Tree 23 yields (sym): -a - b*c**2 + b + c**2 + 1/c  
Tree 24 yields (sym): a/c - b**2 - 3*b - c  
Tree 25 yields (sym): -a**2/c + a*b*c - a - b + 2*c  
Tree 26 yields (sym): -a**2*c**2 - a*c/b + 2*b + 3*c + 2  
Tree 27 yields (sym): -a*b - a*c**2 - b + b/c  
Tree 28 yields (sym): a*b**2 + 2*a - a/b + b*c - c + c/a
```