

# Self-Driving AI Car Simulation

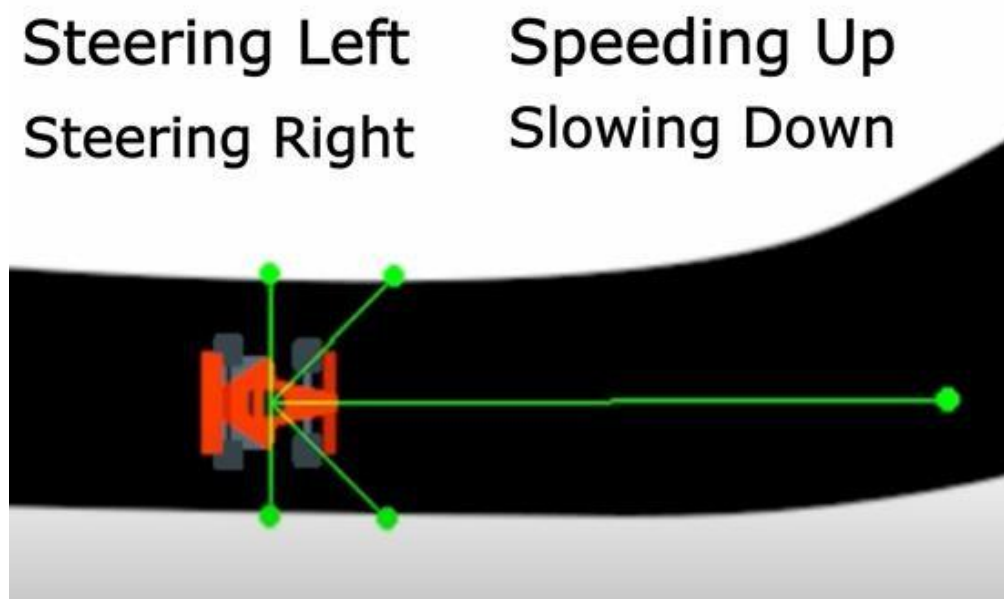
## About the project:

A self-driving car, also known as an autonomous vehicle, a driverless car or a robotic car which can perceive its surroundings and working without the need for human intervention.

A self-driving car is the most attention-grabbing application of artificial intelligence and a classic example of Reinforcement Learning.

Reinforcement learning is an area of machine learning. It's all about taking the right steps to get the most out of a given scenario. Various applications and robots use it to determine the best potential action or path in each situation.

We all know self-driving cars is one of the hottest areas of research and business for the tech giants. What seemed like a science-fiction, a few years ago, now seems more like something which is soon to become a part and parcel of life. The reason of saying "soon to be" is because even though companies like Tesla, Nissan, Cadillac do have self-driving car assistance software, but they still require a human to keep an eye on the road and take control when needed. However, it is fascinating to see how far we have come in terms of innovation and how fast technology is advancing. So much so, that now, with the help of basic deep learning, neural network magic, we can build our own pipeline for autonomous driving!



## TECHNOLOGIES:

### PYTHON

Python is a programming language that lets you work more quickly and integrate your systems more effectively.

Python has a standard library in development, and a few for AI. It has an intuitive syntax, basic control flow, and data structures. It also supports interpretive run-time, without standard compiler languages. This makes Python especially useful for **prototyping algorithms for AI**.

Python offers the least code among others and is in fact 1/5 the number compared to other OOP languages. No wonder it is one of the most popular in the market today.

- Python has Prebuilt Libraries like NumPy for scientific computation, SciPy for advanced computing and Pybrain for machine learning (Python Machine Learning) making it one of the best languages For AI.
- Python developers around the world provide comprehensive support and assistance via forums and tutorials making the job of the coder easier than any other popular languages.
- Python is platform Independent and is hence one of the most flexible and popular choices for use across different platforms and technologies with the least tweaks in basic coding.
- Python is the most flexible of all others with options to choose between OOPs approach and scripting. You can also use IDE itself to check for most codes.

### NEAT

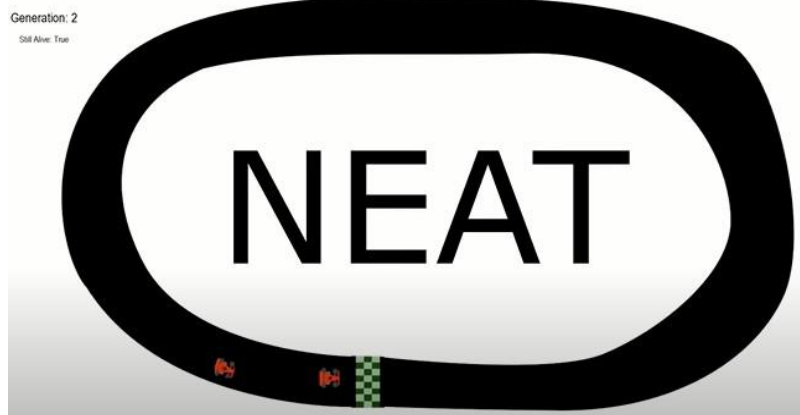
*NEAT* (NeuroEvolution of Augmenting Topologies) is an evolutionary algorithm that creates artificial neural networks . NEAT is a method developed by Kenneth. O. Stanley for evolving arbitrary neural networks. NEAT-Python is a pure Python implementation of NEAT, with no dependencies other than the Python standard library.

In the current implementation of NEAT-Python, a population of individual [genomes](#) is maintained. Each genome contains two sets of [genes](#) that describe how to build an artificial neural network:

1. [Node](#) genes, each of which specifies a single neuron.
2. [Connection](#) genes, each of which specifies a single connection between neurons.

To evolve a solution to a problem, the user must provide a fitness function which computes a single real number indicating the quality of an individual genome: better ability to solve the problem means a higher score. The algorithm progresses through a user-specified number of generations, with each generation being produced by reproduction (either sexual or asexual) and mutation of the most fit individuals of the previous generation.

The reproduction and mutation operations may add nodes and/or connections to genomes, so as the algorithm proceeds genomes (and the neural networks they produce) may become more and more complex. When the pre-set number of generations is reached, or when at least one individual (for a [fitness criterion function](#) of max; others are configurable) exceeds the user-specified [fitness threshold](#), the algorithm terminates.

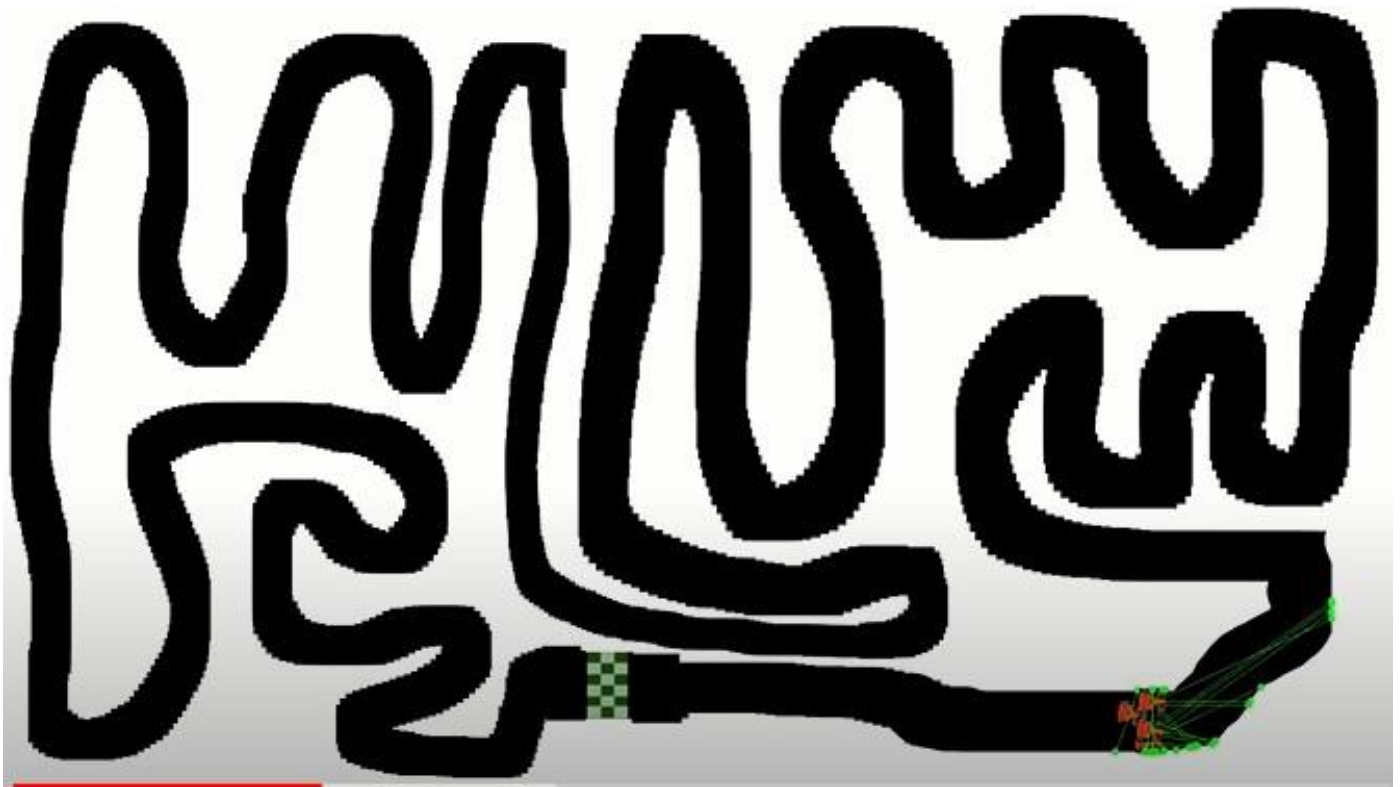


## PURPOSE:

Self-driving cars are a huge milestone not just from the technological standpoint but also from the operational point of view. You see, these vehicles have everything it takes to make our everyday work accelerated and facilitated. With self-driving cars:

- Companies running them can save time and money (e.g., drivers can focus on more complicated work) and even operate 24/7, all year round.
- The number of accidents decreases (AI algorithms are never tired, intoxicated, or sleepy)

Of course, we're not saying here that autonomous vehicles are already in common use. This is still a project in the making, partly due to legal regulations in many countries, which forbid autonomous vehicles from driving on public roads. However, it's just a temporary complication. As technology is growing and becoming more prevalent, the law will have to keep up with these changes.



### **Prerequisites:**

1. This article assumes a basic understanding of Convolutional Neural Networks and its working.
2. The code mentioned here is written in Python using the Pytorch framework so basic knowledge of the language and the framework is recommended.

## Resources for the Project

1. Of course, Python
2. VS Code
3. If your machine does not support GPU, then I would recommend using [Google Colab](#) to train your network. It provides GPU and TPU hours for free
4. Github: <https://github.com/monokim/framework> ...
5. NEAT: <https://neat-python.readthedocs.io/en...>
6. Youtube – Cheesy AI

