How to Build Self-Driving Cars From Scratch

Part 2
Neural Network & Genetic Algorithm

A Step-by-Step Guide to
Creating Autonomous Vehicles With Python

Simulations, scripts, graphics
and other assets included

By
Bolakale Aremu

How to Build Self-Driving Cars From Scratch

A Step-by-Step Guide to Creating Autonomous Vehicles with Python

ISBN: 979-8-3302-3181-2

Published in the United States

Table of Contents

0. Introduction

This is part 2 of my 3-part training guide on how to build self-driving cars from scratch. This guide is bundled with a repository containing simulations, python scripts, graphics, and other useful assets. In this step-by-step guide, I teach you how to make an app that you can use to create a simulation where cars learn how to drive autonomously over racing tracks. Here's a break down of the contents of this guide.

Part 1: **Car mechanics**. In this part, you'll learn how to draw the car and control it with the keyboard. You will use a multimedia library for Python called Pyglet. (https://pyglet.org/). This is the only library you will use in this guide. This is a cross-platform windowing and multimedia library for Python. It's a powerful yet easy-to-use Python library for building games and other visually rich applications on Windows, macOS, and Linux.

Part 2: **Neural network and genetic algorithm**. You'll learn how to create the AI where you combine a neural network and genetic algorithm. You'll learn how to add sensors to the car and get output from them. To prevent the untrained network from car crashes, a genetic algorithm will be used to train the cars. This will help the cars to drive simple tracks.

Part 3: **Challenges**. You'll add some challenges to the system. Tracks get more complicated and will take advantage from the previous track training by storing and retrieving the car brains.

By the end of this training, you will have created self-driving cars that are capable of driving on unknown tracks by understanding how to steer, accelerate, and brake based on what cars see in front of them. Since autonomous cars need a brain of some kind, you know we need some AI (artificial intelligence). AI comes in many forms, but in this guide, you'll use a neural network where the weights are adjusted by a genetic algorithm.

Employment opportunities (https://www.turing.com/jobs) often come from work samples and concrete skills, rather than a college degree. So, you need to learn the practical aspect well enough. This guide will not only help you learn well and build a stunning portfolio, it will also provide you continuous help and support.

With this book and my dedicated 24/7 help and support team, there's nothing for you to fear. I have helped many Python developers update their automation development skills, launch successful careers and get hired for remote jobs (https://vanhack.com/jobs). I notice that even the most ambitious beginners can run into problems, such as unable to decide where to begin. Sometimes they get completely lost on the way and therefore need further help.

In Chapter 13, I explain how to download my repository which contains all updates of the Python scripts (codes) and simulations used in this Part 2 of the guide. Although I explain all the codes used in this guide clearly, if you need further help, just use my support link at the end of the Chapter.

The truth is everyone needs help at one point or the other to learn and build automation in their development journey. I can give you more challenges and their solutions in my subsequent trainings.

0.1. Getting Started

How does it work? Well, first of all, **no AI libraries** will be used in the training. That means we will build everything from scratch. For this, the following will help you have beginner experience in Python or another programming language.

1. You have basic math skills.
2. You have a keen interest in artificial intelligence.
3. You have a minimum of **Python 3.12** installed on your computer.

If your Python skills or math skills are a bit rusty, don't worry. Even with basic Python knowledge, you should be able to follow along and learn lot from the course. So, to get started, go to Chapter 3 to download the codes and other assets you need for this training.

0.2. The Target Audience

Students who will benefit the most from this training are

1. Developers who want to use their basic Python skills to program self-driving cars.
2. Developers who want to understand neural networks and genetic algorithms by building them from the ground up.

There are no exercises in the course. This is a tutorial bundled with simulations and Python scripts that are available in my repository for students who legally purchased this book. However, I strongly recommend that you type your code along with me. At certain sections of this guide, references are made to the right simulations you should see to check your progress.

0.3. How to Use This Guide

I suggest coding along with me as we go through the material. Alternatively, you can first thoroughly review each section of a chapter before attempting to write the code on your own. Once you're confident you understand a section, try writing the code yourself. Afterward, compare your work to mine. You can do this by locating and opening the relevant files (such as canvas.py, car.py, and training.py) mentioned in the section. Use Visual Studio or any IDE of your choice to open these files and assess how your code measures up against the examples provided.

0.4. Who am I?

My name is Bolakale Aremu, a software engineer and trainer. I teach beginners how to create their first programs and I help professionals to understand Python better. I have worked for many companies including Philips and Microsoft. During my many years of experience, I have always combined software engineering with training on the job, and now you are going to benefit from this as well.

So, are you ready to dive in to the introduction to neural networks and genetic algorithms? If yes, I wish you good luck!

0.5. The Tool You Need

First you must download and install Python 3.12.2 or higher (https://www.python.org/downloads/). I highly recommend you also install Visual Studio editor (https://code.visualstudio.com/download). Alternatively. You can install Pycharm IDE (https://www.jetbrains.com/pycharm/download/) right now to get started. This is where you will write your codes.

They are very easy to install and have a bunch of developer tools pre-installed. There are tons of videos on Youtube on how to install and use the latest versions of Visual Studio and Pycharm. Once you've installed one of these IDEs, you're good to go!

1. First Track

The first track looks like Figure 1.1.

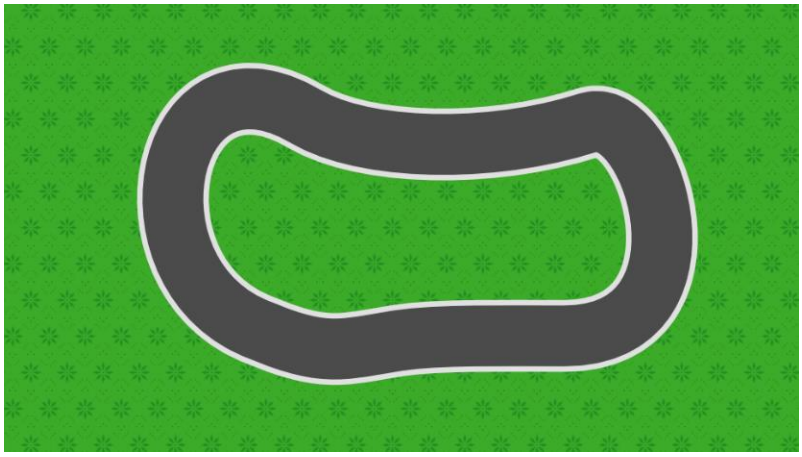

Figure 1.1: The first track for our cars to practice

The assets for this track are found in the folder named *assets-track0*. The name of this track is *track0.png*. But what are the other two files in this folder?

Well, I thought that our track background looks a bit dull, so I decided to give each track an overlay file just to make it a bit more exciting. Figure 1.2 shows the *track0.png* track with the overlay on top.

Figure 1.2: Track0 with the overlay on top

You might wonder why each track has separate files for the background and overlay. The reason is that I want certain elements to be on top of the car, so it looks like the car is driving underneath them. See Figure 1.3.

Figure 1.3: Track0 with the overlay and a car under

This effect can only be created if the overlay is drawn separately from the background. In the *assets-track0* folder, there is also the Json file. See Figure 1.4.

Figure 1.4: The Json file

The Json file contains a key with a list of checkpoints. For now, the list contains a single coordinate, and it will be the starting point for the cars.

Now go to Chapter 13 to download the repository for this Part 2. The folder named *finished-code-06* contains all the three assets mentioned above. It also contains the already completed code up to the end of this chapter.