Self-Driving Car Nanodegree Syllabus

Contact Info

While going through the program, if you have questions about anything, you can reach us at enterprise-support@udacity.com. For help from Udacity Mentors and your peers visit the Udacity Classroom.

Nanodegree Program Info

Version: 1.0.0

Length of Program: 271 Days*

** This is a self-paced program and the length is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. Actual hours may vary.*

Part 1: Computer Vision and Deep Learning

In this term, you'll become an expert in applying Computer Vision and Deep Learning on automotive problems. You will teach the car to detect lane lines, predict steering angle, and more all based on just camera data!

Project: Finding Lane Lines on the Road

In the first project, apply your new computer vision skills to find lane markings on the road!

Supporting Lessons

Lesson	Summary
Welcome	In this lesson, we'll introduce you to the program, help you discover the services we provide, and show you all the incredible projects you'll build. Get ready for an incredible journey!
Computer Vision Fundamentals	In this first lesson, you'll discover some basic computer vision techniques to find lane markings on the road. We'll dive much deeper into computer vision in later lessons, so relax and have fun!

Project: Traffic Sign Classifier

You just finished getting your feet wet with deep learning. Now put your skills to the test by using deep learning to build your own model to classify different traffic signs!

Supporting Lessons

Lesson	Summary
Introduction to Neural Networks	Learn to build and train neural networks, starting with the foundations in linear and logistic regression, and culminating in backpropagation and multilayer perceptron networks.
MiniFlow	In this assignment, you will build your own neural network library from scratch! Your library, MiniFlow, will behave much like TensorFlow, Google's deep learning library.
Introduction to TensorFlow	Vincent Vanhoucke, Principal Scientist at Google Brain, introduces you to deep learning and Tensorflow, Google's deep learning framework.
Deep Neural Networks	Vincent walks you through how to go from a simple neural network to a deep neural network. You'll learn about why additional layers can help and how to prevent overfitting.
Convolutional Neural Networks	Vincent explains the theory behind Convolutional Neural Networks and how they help us dramatically improve performance in image classification.
LeNet for Traffic Signs	Using the infamous LeNet neural network architecture, take your first steps toward building a Traffic Sign classifier!

Project: Behavioral Cloning

Put your deep learning skills to the test with this project! Train a deep neural network to drive a car like you!

Supporting Lessons

Lesson	Summary
Keras	Take on the neural network framework, Keras. You'll be amazed how few lines of code you'll need to build and train deep neural networks!
Transfer Learning	Learn about some of the most famous neural network architectures and how you can use them. By the end of this lesson, you'll know how to create new models by leveraging existing canonical networks.

Project: Advanced Lane Finding

Build a pipeline using distortion correction, image rectification, color transforms, and gradient thresholding to identify lane lines and their curvature in a video.

Supporting Lessons

Lesson	Summary
Advanced Techniques for Lane Finding	Discover more advanced computer vision techniques to improve upon your lane lines algorithm!

Project: Vehicle Detection and Tracking

In the final Term 1 project, you'll leverage your object detection abilities toward a video pipeline for vehicle detection and tracking.

Supporting Lessons

Lesson	Summary
Machine Learning and Stanley	Learn about classification, training and testing, and run a naive Bayes classifier using Scikit Learn.
Support Vector Machines	See how support vector machines work, and code one using Scikit Learn.
Decision Trees	Learn about decision trees, including entropy and information gain.
Object Detection	In this lesson, you'll learn how to detect and track vehicles using color and gradient features and a support vector machine classifier.

Part 2: Sensor Fusion, Localization, and Control

In this term, you'll learn how to use an array of sensor data to perceive the environment and control the vehicle. You'll evaluate sensor data from camera, radar, lidar, and GPS, and use these in closed-loop controllers that actuate the vehicle.

Project: Extended Kalman Filters

In this project, you'll apply everything you've learned so far about Sensor Fusion by implementing an Extended Kalman Filter in C++!

Supporting Lessons

Lesson	Summary
Introduction and Sensors	Meet the team at Mercedes who will help you track objects in real-time with Sensor Fusion.
Kalman Filters	Learn from the best! Sebastian Thrun will walk you through the usage and concepts of a Kalman Filter using Python.
C++ Checkpoint	Are you ready to build Kalman Filters with C++? Take these quizzes to find out.
Two-Dimensional Robot Motion and Trigonometry	Learn the basics of trigonometry and how to decompose a self driving car's motion into X and Y components.
Lidar and Radar Fusion with Kalman Filters in C++	In this lesson, you'll build a Kalman Filter in C++ that's capable of handling data from multiple sources. Why C++? Its performance enables the application of object tracking with a Kalman Filter in r

Project: Unscented Kalman Filters

Put your skills to the test! Use C++ to code an Unscented Kalman Filter capable of tracking non-linear motion.

Supporting Lessons

Lesson	Summary
Unscented Kalman Filters	While Extended Kalman Filters work great for linear motion, real objects rarely move linearly. With Unscented Kalman Filters, you'll be able to accurately track non-linear motion!

Project: Kidnapped Vehicle

In this project, you'll build a particle filter and combine it with a real map to localize a vehicle!

Supporting Lessons

Lesson	Summary
Introduction to Localization	Meet the team that will guide you through the localization lessons!
Localization Overview	Sebastian Thrun will give you an overview of the theory behind localization!
Markov Localization	In this lesson, you'll learn the math behind localization as well as how to implement Markov localization in C++.
Motion Models	Here you'll learn about vehicle movement and motion models to predict where your car will be at a future time.
Particle Filters	Now, Sebastian will teach you what a particle filter is and will teach the theory and math behind the particle filter.
Implementation of a Particle Filter	Now that you understand how a particle filter works, you'll learn how to code a particle filter.

Project: PID Controller

Implement a PID Controller to race around the lake track.

Supporting Lessons

Lesson	Summary
PID Control	In this lesson, you'll learn about and how to use PID controllers with Sebastian!

Project: Model Predictive Control

In this project, you'll implement Model Predictive Control to drive the vehicle around the track even with additional latency between commands!

Supporting Lessons

Lesson	Summary
Vehicle Models	In this lesson, you'll learn about kinematic and dynamic vehicle models. We'll use these later with Model Predictive Control.
Model Predictive Control	In this lesson, you'll learn how to frame the control problem as an optimization problem over time horizons. This is Model Predictive Control!

Part 3: Path Planning, Concentrations, and Systems

In this term, you'll learn how to plan where the vehicle should go, how the vehicle systems work together to get it there, and you'll perform a deep-dive into a concentration of your choice.

Project: Path Planning Project

Drive a car down a highway with other cars using your own path planner.

Supporting Lessons

Lesson	Summary
Search	In this lesson, you will learn about discrete path planning and algorithms for solving the path planning problem.
Prediction	In this lesson, you'll learn how to use data from sensor fusion to generate predictions about the likely behavior of moving objects.
Behavior Planning	In this lesson, you'll learn how to think about high level behavior planning in a self driving car.
Trajectory Generation	In this lesson, you'll use C++ and the Eigen linear algebra library to build candidate trajectories for the vehicle to follow.

Project: Semantic Segmentation

Train segmentation networks, which paint each pixel of the image a different color, based on its class. Use segmented images to find free space on the road.

Supporting Lessons

Lesson	Summary
Elective: Advanced Deep Learning	Students in this specialization, built with the NVIDIA Deep Learning Institute, will learn about semantic segmentation, and inference optimization, active areas of deep learning research.
Fully Convolutional Networks	In this lesson, you'll learn the motivation for Fully Convolutional Networks and how they are structured.
Scene Understanding	In this lesson, you'll be introduced to the problem of Scene Understanding and the role FCNs play.
Inference Performance	In this lesson, you'll become familiar with various optimizations in an effort to squeeze every last bit of performance at inference.

Project: Functional Safety of a Lane Assistance System

It's time to start the functional safety project! You will be creating a safety case for a lane assistance system.

Supporting Lessons

Lesson	Summary
Elective: Functional Safety	If you choose the Functional Safety specialization, built with Elektrobit, you’ll learn functional safety frameworks to ensure that vehicles are safe, both at the system and component levels.
Introduction to Functional Safety	You will learn to make safer vehicles using risk evaluation and systems engineering.
Safety Plan	A functional safety plan is critical to any functional safety project. Here you will learn what goes into a safety plan so that you can document your own.
Hazard Analysis and Risk Assessment	In a hazard analysis and risk assessment, you will identify vehicular malfunctions and evaluate their risk levels. You can then derive safety goals defining how your vehicle will remain safe.
Functional Safety Concept	You will derive functional safety requirements from the safety goals and then add extra functionality to the system diagram. Finally, you will document your work, a part of functional safety.
Technical Safety Concept	Once you have derived functional safety requirements, you drill down into more detail. In the technical safety concept, you refine your requirements into technical safety requirements.
Functional Safety at the Software and Hardware Levels	The last step in the vehicle safety design phase is to derive hardware and software safety requirements. In this lesson, you will derive these requirements and refine a software system architecture.

Project: Programming a Real Self-Driving Car

Run your code on Carla, Udacity's autonomous vehicle!

Supporting Lessons

Lesson	Summary
Autonomous Vehicle Architecture	Learn about the system architecture for Carla, Udacity's autonomous vehicle.
Introduction to ROS	Obtain an architectural overview of the Robot Operating System Framework and setup your own ROS environment on your computer.
Packages & Catkin Workspaces	Learn about ROS workspace structure, essential command line utilities, and how to manage software packages within a project. Harnessing these will be key to building shippable software using ROS.
Writing ROS Nodes	ROS Nodes are a key abstraction that allows a robot system to be built modularly. In this lesson, you'll learn how to write them using Python.

Udacity

Generated Wed Feb 19 08:36:35 PST 2020