UDACITY

NANODEGREE PROGRAM SYLLABUS

Deep Learning

Overview

The Deep Learning Nanodegree program offers you a solid introduction to the world of artificial intelligence. In this program, you'll master fundamentals that will enable you to go further in the field, launch or advance a career, and join the next generation of deep learning talent that will help define a beneficial, new, AI-powered future for our world. You will study cutting-edge topics such as Neural Networks, Convolutional Neural Networks, Recurrent Neural Networks, Generative Adversarial Networks, and Network Deployment, and build projects in PyTorch and NumPy. You'll learn from authorities such Ian Goodfellow and Jun-Yan Zhu, inventors of types of generative adversarial networks, as well as AI experts, Sebastian Thrun and Andrew Trask. For anyone interested in this transformational technology, this program is an ideal point-of-entry.

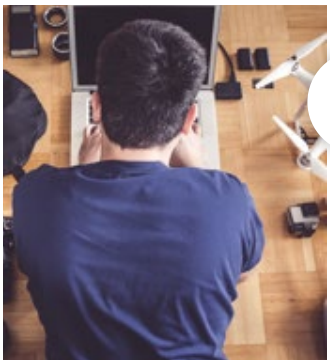
The program is comprised of 5 courses and 5 projects. Each project you build will be an opportunity to prove your skills and demonstrate what you've learned in your lessons.

IN COLLABORATION WITH

aws

facebook Artificial Intelligence

Estimated Time:
4 Months at
12 hrs/week



Prerequisites:
Basic Python

Flexible Learning:
Self-paced, so you can learn on the schedule that works best for you

Need Help?
udacity.com/advisor
Discuss this program with an enrollment advisor.

Course 1: Neural Networks

Learn neural networks basics, and build your first network with Python and NumPy. Use the modern deep learning framework PyTorch to build multi-layer neural networks, and analyze real data.

Course Project
Predicting Bike-Sharing Patterns

Learn neural networks basics, and build your first network with Python and NumPy. You'll define and train a multi-layer neural network, and use it to analyze real data. In this project, you will build and train neural networks from scratch to predict the number of bike-share users on a given day.

	LEARNING OUTCOMES	
LESSON ONE	**Introduction to Neural Networks**	• In this lesson, you will learn solid foundations on deep learning and neural networks. You'll also implement gradient descent and backpropagation in Python.
LESSON TWO	**Implementing Gradient Descent**	• Mat and Luis will introduce you to a different error function and guide you through implementing gradient descent using NumPy matrix multiplication..
LESSON THREE	**Training Neural Networks**	• Now that you know what neural networks are, in this lesson, you will learn several techniques to improve their training. Learn how to prevent overfitting of training data and best practices for minimizing the error of a network.
LESSON FOUR	**Sentiment Analysis**	• In this lesson, Andrew Trask, the author of Grokking Deep Learning, will show you how to define and train a neural networks for sentiment analysis (identifying and categorizing opinions expressed in text).
LESSON FIVE	**Deep Learning WIth Pytorch**	• Learn how to use PyTorch for building and testing deep learning models

Course 2: Convolutional Neural Networks

Learn how to build convolutional networks and use them to classify images (faces, melanomas, etc.) based on patterns and objects that appear in them. Use these networks to learn data compression and image denoising.

Course Project
Landmark Classification & Tagging for Social Media

In this project, you will apply the skills you have acquired in the course to build a landmark classifier. Photo-sharing services or photo-storage services may use landmark classification to automatically tag photos with relevant hashtags or location markers. This type of functionality could be especially important when photo location metadata is not available, which could happen when a photo is taken without metadata (e.g., phone was on airplane mode, camera was old and without GPS) or if a photo has had its metadata scrubbed. In the project, you will go through a machine learning design process end-to-end: performing data preprocessing and augmentation, designing your own CNN from scratch, and training and saving your best CNN model. You will also use transfer learning and compare your transfer-learned model with your from-scratch CNN.

	LEARNING OUTCOMES	
LESSON ONE	**Cloud Computing**	• Take advantage of Amazon's GPUs to train your neural network faster. In this lesson, you'll setup an instance on AWS and train a neural network on a GPU.
LESSON TWO	**Convolutional Neural Network**	• Alexis and Cezanne explain how Convolutional Neural Networks can be used to identify patterns in images and how they help us dramatically improve performance in image classification tasks.
LESSON THREE	**CNNs In PyTorch**	• In this lesson, you'll walk through an example Convolutional Neural Network (CNN) in PyTorch. You'll study the line-by-line breakdown of the code and can download the code and run it yourself.

LESSON FOUR	**Weight Initialization**	• In this lesson, you'll learn how to find good initial weights for a neural network. Having good initial weights often allows a neural network to arrive at an optimal solution, faster than without initialization.
LESSON FIVE	**Autoencoders**	• Autoencoders are neural networks used for data compression, image denoising, and dimensionality reduction. Here, you'll build autoencoders using PyTorch.
LESSON SIX	**Transfer Learning in PyTorch**	• Most people don't train their own networks on massive datasets. In this lesson, you'll learn how to finetune and use a pretrained network and apply it to a new task using transfer learning.
LESSON SEVEN	**Deep Learning for Cancer Detection**	• In this lesson, Sebastian Thrun teaches us about his groundbreaking work detecting skin cancer with Convolutional Neural Networks.

Course 3: Recurrent Neural Networks

Build your own recurrent networks and long short-term memory networks with PyTorch; perform sentiment analysis and use recurrent networks to generate new text from TV scripts.

Course Project
Generate TV Scripts

In this project, you will build your own Recurrent Networks and Long Short-Term Memory Networks with PyTorch. You'll perform sentiment analysis and generate new text, and use recurrent networks to generate new text that resembles a training set of TV scripts.

	LEARNING OUTCOMES	
LESSON ONE	**Recurrent Neural Networks**	• Ortal will introduce Recurrent Neural Networks (RNNs), which are machine learning models that are able to recognize and act on sequences of inputs.
LESSON TWO	**Long Short-Term Memory Network**	• Luis explains Long Short-Term Memory Networks (LSTM), and similar architectures that form a memory about a sequence of inputs, over time.
LESSON THREE	**Implementation of RNN & LSTM**	• Train recurrent neural networks to generate new characters,words, and bodies of text.
LESSON FOUR	**Hyperparameters**	• In this lesson, we'll look at a number of different hyperparameters that are important for our deep learning work, such as learning rates. We'll discuss starting values and intuitions for tuning each hyperparameter.
LESSON FIVE	**Embeddings & Word2vec**	• In this lesson, you'll learn about embeddings in neural networks by implementing a word2vec model that converts words into a representative vector of numerical values.
LESSON SIX	**Sentiment Prediction RNN**	• In this lesson, you'll learn to implement a recurrent neural network for predicting sentiment. This is intended to give you more experience building RNNs.

Course 4: Generative Adversarial Networks

Learn to understand and implement a Deep Convolutional GAN (generative adversarial network) to generate realistic images, with Ian Goodfellow, the inventor of GANs, and Jun-Yan Zhu, the creator of CycleGANs.

Course Project
Generate Faces

Learn to understand Generative Adversarial Networks with the model's inventor, Ian Goodfellow. Then, apply what you've learned in this project and implement a Deep Convolutional GAN. This DCGAN is made of a pair of multi-layer neural networks that compete against each other until one learns to generate realistic images of faces.

	LEARNING OUTCOMES	
LESSON ONE	**Generative Adversarial Network**	• Ian Goodfellow, the inventor of GANs, introduces you to these exciting models. You'll also implement your own GAN on a simple dataset.
LESSON TWO	**Deep Convolutional GANs**	• Implement a Deep Convolutional GAN to generate complex, color images of house numbers.
LESSON THREE	**PIX2PIX & Cyclegan**	• Jun-Yan Zhu and Cezanne lead you through a CycleGAN formulation that can learn from unlabeled sets of images.

Course 5: Updating a Model

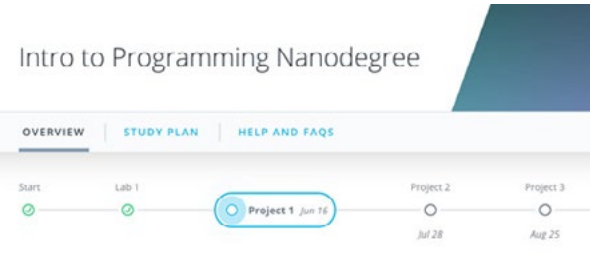
In this project, you will train and deploy your own PyTorch sentiment analysis model using Amazon SageMaker on AWS. This model will be trained to do sentiment analysis on movie reviews (positive or negative reviews). You'll build the model, deploy it, and create a gateway for accessing this model from a website.

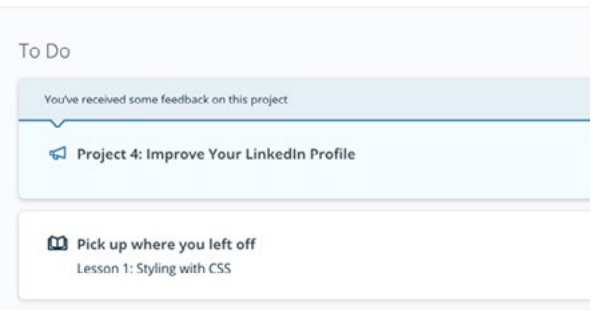
Course Project
Deploying a Sentiment Analysis Model

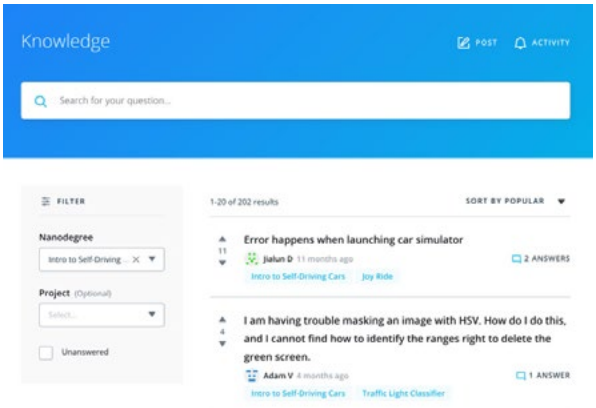
In this project, you will train and deploy your own PyTorch sentiment analysis model using Amazon SageMaker on AWS. This model will be trained to do sentiment analysis on movie reviews (positive or negative reviews). You'll build the model, deploy it, and create a gateway for accessing this model from a website.

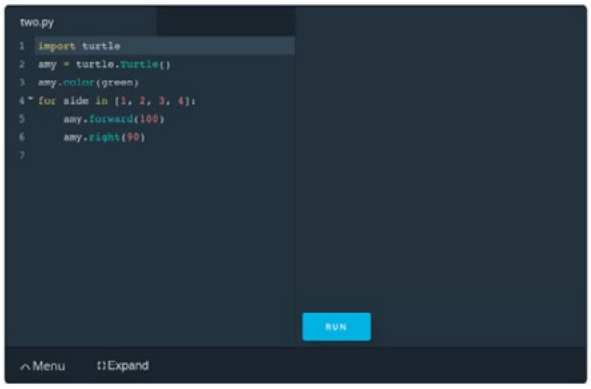
	LEARNING OUTCOMES	
LESSON ONE	**Introduction to Deployment**	• Learn where cloud deployment is used in industry and about various methods for deployment (websites, apps, etc.). Become familiar with cloud deployment terminology.
LESSON TWO	**Deploy a Model**	• Deploy a model using Amazon SageMaker and learn to apply built-in algorithms, like XGBoost, to a variety of tasks.
LESSON THREE	**Custom Models & Web Hosting**	• In this lesson, you'll train and deploy your own PyTorch model. Then, see how to define a gateway using SageMaker to allow for outside-access to your model. See how your model responds to user input.
LESSON FOUR	**Model Monitoring**	• In this lesson, learn how to interpret log messages and monitor the behavior of your model over time. See how to implement an A/B test, in SageMaker, to evaluate the performance of two different models.
LESSON FIVE	**Updating a Model**	• Developing a machine learning model is an iterative process. Learn how to look at indicators like data distribution to see if you should update a model.

Our Classroom Experience









REAL-WORLD PROJECTS

Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

KNOWLEDGE

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover in real-time how to solve the challenges that you encounter.

STUDENT HUB

Leverage the power of community through a simple, yet powerful chat interface built within the classroom. Use Student Hub to connect with fellow students in your program as you support and learn from each other.

WORKSPACES

See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

QUIZZES

Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

CUSTOM STUDY PLANS

Preschedule your study times and save them to your personal calendar to create a custom study plan. Program regular reminders to keep track of your progress toward your goals and completion of your program.

PROGRESS TRACKER

Stay on track to complete your Nanodegree program with useful milestone reminders.

Learn with the Best

Mat Leonard

INSTRUCTOR

Mat is a former physicist, research neuroscientist, and data scientist. He did his PhD and Postdoctoral Fellowship at the University of California, Berkeley.

Luis Serrano

INSTRUCTOR

Luis was formerly a Machine Learning Engineer at Google. He holds a PhD in mathematics from the University of Michigan, and a Postdoctoral Fellowship at the University of Quebec at Montreal.

Cezanne Camacho

CURRICULUM LEAD

Cezanne is a computer vision expert with a Masters in Electrical Engineering from Stanford University. As a former genomics and biomedical imaging researcher, she's applied computer vision and deep learning to medical diagnostics.

Alexis Cook

INSTRUCTOR

Alexis is an applied mathematician with a Masters in computer science from Brown University and a Masters in applied mathematics from the University of Michigan. She was formerly a National Science Foundation Graduate Research Fellow.

Learn with the Best

Jennifer Staab

INSTRUCTOR

Jennifer has a PhD in Computer Science and a Masters in Biostatistics; she was a professor at Florida Polytechnic University. She previously worked at RTI International and United Therapeutics as a statistician and computer scientist.

Sean Carrell

INSTRUCTOR

Sean Carrell is a former research mathematician specializing in Algebraic Combinatorics. He completed his PhD and Postdoctoral Fellowship at the University of Waterloo, Canada.

Ortal Arel

INSTRUCTOR

Ortal Arel is a former computer engineering professor. She holds a Ph.D. in Computer Engineering from the University of Tennessee. Her doctoral research work was in the area of applied cryptography.

Jay Alammar

INSTRUCTOR

Jay has a degree in computer science, loves visualizing machine learning concepts, and is the Investment Principal at STV, a $500 million venture capital fund focused on high-technology startups.

Learn with the Best

Daniel Jiang

INSTRUCTOR

Daniel is a machine learning engineer who studied computer science at the University of California, Berkeley. He has worked on machine learning research at a variety of industry and academic groups.

All Our Nanodegree Programs Include:

EXPERIENCED PROJECT REVIEWERS

REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve

TECHNICAL MENTOR SUPPORT

MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions

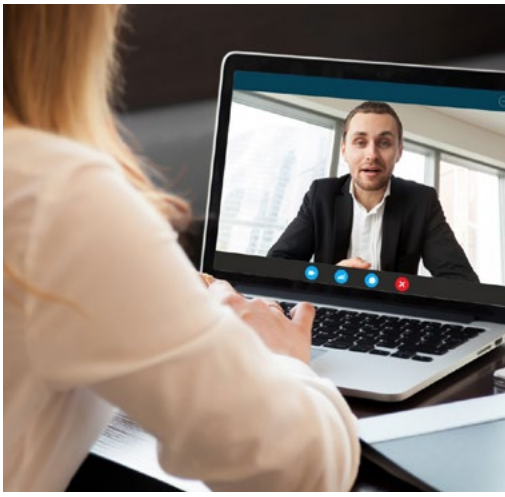
PERSONAL CAREER SERVICES

CAREER SUPPORT

- Resume support
- Github portfolio review
- LinkedIn profile optimization

Frequently Asked Questions

PROGRAM OVERVIEW



WHY SHOULD I ENROLL?

In this program, you'll master deep learning fundamentals that will prepare you to launch or advance a career, and additionally pursue further advanced studies in the field of artificial intelligence. You will study cutting-edge topics such as neural, convolutional, recurrent neural, and generative adversarial networks, as well as sentiment analysis model deployment. You will build projects in NumPy and PyTorch. You will learn from experts in the field, and gain exclusive insights from working professionals. For anyone interested in building expertise with this transformational technology, this Nanodegree program is an ideal point-of-entry.

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

This program is designed to build on your skills in deep learning. As such, it doesn't prepare you for a specific job, but expands your skills in the deep learning domain. These skills can be applied to various applications and also qualify you to pursue further studies in the field.

HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

If you are interested in the fields of artificial intelligence and machine learning, this Nanodegree program is the perfect way to get started!

ENROLLMENT AND ADMISSION

DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

No. This Nanodegree program accepts all applicants regardless of experience and specific background.

WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

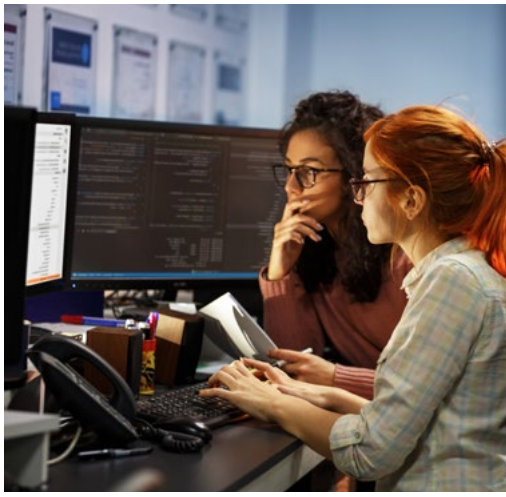
Students who are interested in enrolling must have intermediate-level Python programming knowledge, and experience with NumPy and pandas. You will need to be able to communicate fluently and professionally in written and spoken English. Additionally, students must have the necessary math knowledge, including: algebra and some calculus—specifically partial derivatives, and matrix multiplication (linear algebra).

IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

We have a number of Nanodegree programs and free courses that can help you prepare, including:

- Introduction to Data Analysis
- Introduction to Computer Science
- Introduction to Python
- Linear Algebra Refresher
- Introduction to Programming Nanodegree program with Data Analysis specialization

FAQs Continued



TUITION AND TERM OF PROGRAM

HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The Deep Learning Nanodegree program is comprised of content and curriculum to support five (5) projects. We estimate that students can complete the program in four (4) months working 10 hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided and if you do not pass the project, you will be asked to resubmit the project until it passes.

HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified in the payment card above. If you do not graduate within that time period, you will continue learning with month to month payments. See the **Terms of Use** and **FAQs** for other policies regarding the terms of access to our Nanodegree programs.

I HAVE GRADUATED FROM THE DEEP LEARNING NANODEGREE PROGRAM BUT I WANT TO KEEP LEARNING. WHERE SHOULD I GO FROM HERE?

Graduates from this Nanodegree program earn guaranteed admitted status into our more advanced Self-Driving Car Engineer or Flying Car Nanodegree programs, subject to payment by student for the cost of enrollment for those Nanodegree programs.

SOFTWARE AND HARDWARE

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

Virtually any 64-bit operating with at least 8GB of RAM will be suitable. Students should also have Python 3 and Jupyter Notebooks installed. For the more intensive portions of the program that come later, we will be providing students with AWS instances where geographically possible.