INDIVIDUAL LEARNERS

UDACITY

SCHOOL OF ARTIFICIAL INTELLIGENCE

Deep Learning

Nanodegree Program Syllabus

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 or launch a brand new career. You will study cutting-edge topics such as neural networks, convolutional neural networks, recurrent neural networks, and generative adversarial networks. Plus, you'll build projects in PyTorch. Join the next generation of deep learning talent that will help define a highly beneficial AI-powered future for our world.

Learning Objectives

A graduate of this program will be able to:

- Create a simple neural network (NN) using PyTorch to predict patterns from real data.
- Build convolutional neural networks (CNNs) to classify landmark images based on patterns and objects that appear in them.
- Implement a recurrent neural network (RNN) and its variants (LSTMs, GRUs) with PyTorch to build a customer service chatbot.

Program information

Estimated Time

4 months at 10hrs/week*

Skill Level

Intermediate

Prerequisites

Students should have experience working with and/or knowledge of the following topics:

- Derivatives
- Linear Algebra
- Numpy, Pandas
- Intermediate Python
- Jupyter notebooks

Required Hardware/Software

NLTK, SKLearn, BeautifulSoup, Numpy

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Course 1

Introduction to Deep Learning

This course covers the fundamental theoretical and practical topics in deep learning. You'll begin by learning about how experts think about deep learning, when it is appropriate to use deep learning, and how to apply the skill. You'll then learn the foundational algorithms underpinning modern deep learning: gradient descent and backpropagation. Once those foundations are established, explore design constructs of neural networks and the impact of these design decisions. Finally, the course explores how neural network training can be optimized for accuracy and robustness using training techniques like early stopping, dropout, regularization, and momentum. Throughout the course, theory and fundamental implementations are woven together with PyTorch code to reinforce both the theory and practice of deep learning.

Course Project

Developing a Handwritten Digits Classifier with PyTorch

In this project, you will develop a handwritten digit recognition system in PyTorch. Then, use data preprocessing skills to load data appropriately for use in models. Develop a neural network using PyTorch and write a training loop that trains the model with the loaded data. Lastly, apply advanced training techniques to improve accuracy on the test set.

Lesson 1

Deep Learning

- Explain the difference between artificial intelligence, machine learning, and deep learning.
- Recognize the power of deep learning by reviewing popular examples of deep learning applications.

Lesson 2

Minimizing the Error Function with Gradient Descent

- Use PyTorch to preprocess data.
- Use maximum likelihood, cross-entropy, and probability to measure model performance.
- Apply gradient descent to minimize error.
- Implement a backpropagation algorithm.
- Identify key components of perceptrons.

Lesson 3

Introduction to Neural Networks

- Explain essential concepts in neural networks.
- Design neural network architectures.
- Distinguish between problems based on the objective of the model.
- Implement appropriate architectures for model objectives.

Lesson 4

Training Neural Networks

- Define a loss function and optimization method to train a neural network.
- Distinguish between overfitting and underfitting, and identify the causes of each.
- Optimize the training process using early stopping, regularication, dropout, learning rate decay, and momentum.
- Distinguish between batch and stochastic gradient descent.
- Build a neural network with PyTorch and run data through it.
- Test and validate a trained network to ensure it generalizes.

Course 2

Convolutional Neural Networks

This course introduces convolutional neural networks, the most widely used type of neural networks specialized in image processing. You will learn the main characteristics of CNNs that make them better than standard neural networks for image processing. Then you'll examine the inner workings of CNNs, including how to build CNNs from scratch to complete image classification tasks. After, you'll apply these architectures to custom datasets using transfer learning and learn to use autoencoders for anomaly detection as well as image denoising. Finally, you will learn how to use CNNs for object detection and semantic segmentation.

Course Project

Landmark Classification and Tagging for Social Media

Photo sharing and photo storage services like to have location data for each photo that is uploaded. With the location data, these services can build advanced features, such as automatic suggestion of relevant tags or automatic photo organization, which help provide a compelling user experience. Although a photo's location can often be obtained by looking at the photo's metadata, many photos uploaded to these services will not have location metadata available. This can happen when, for example, the camera capturing the picture does not have GPS or if a photo's metadata is scrubbed due to privacy concerns.

If no location metadata for an image is available, one way to infer the location is to detect and classify a discernible landmark in the image. Given the large number of landmarks across the world and the immense volume of images that are uploaded to photo sharing services, using human judgment to classify these landmarks would not be feasible.

In this project, you will build models to automatically predict the location of the image based on any landmarks depicted in the image. You will go through the machine learning design process end-to-end: performing data preprocessing, designing and training CNNs, comparing the accuracy of different CNNs, and deploying an app based on the best CNN you trained.

Lesson 1

Introduction to CNNs

- List main applications of CNNs.
- Understand professional roles involved in the development of a CNN-based application.
- Understand the main events in the history of CNNs.

Lesson 2

CNN Concepts

- Recap training networks in PyTorch.
- Use multi-layer perceptron for image classification.
- Understand limitations of MLPs when applied to images.
- Learn the basic concepts of CNNs that make them great at tasks involving images.

Lesson 3

CNNs in Depth

- Learn how to use the basic layers used in CNNs.
- Put all layers together to build a CNN from scratch.
- Classify images using a CNN built from scratch.
- Improve the performances of your CNN.
- Export a model for production.

Lesson 4

Transfer Learning

- Understand key CNN architectures and their innovations.
- Apply multiple ways of adapting pre-trained networks using transfer learning.
- Fine-tune a pre-trained network on a new dataset.

Lesson 5

Autoencoders

- Understand linear and CNN-based autoencoders.
- Design and train a linear autoencoder for anomaly detection.
- Design and train a CNN autoencoder for anomaly detection and image denoising.

Lesson 6

Object Detection and Segmentation

- Understand the architecture of an object detection model.
- Train and evaluate an object detection model.
- Understand the architecture of a semantic segmentation model.
- Train and evaluate a semantic segmentation model.

Course 3

RNNs & Transformers

This course covers multiple RNN architectures and discusses design patterns for those models. Additionally, you'll focus on the latest transformer architectures.

Course Project

LSTM Seq2Seq Chatbot

In this project, learn how to build an AI chatbot using LSTMs, Seq2Seq, and word embeddings for increased accuracy. You'll use a dataset of conversational dialogue to replicate conversational ability. Complete the following steps: Write a neural network architecture using Pytorch, train it with the dataset, and tune network hyperparameters for increased accuracy. In the end, you'll demonstrate your proficiency in deep learning to prospective employers by conversing with their chatbot at the command line.

Lesson 1

Recurrent Neural Networks

- Explain how RNNs evolved from feedforward neural networks.
- Recognize the benefit of RNNs by reviewing the applications of RNNs in areas like machine translation.
- Perform backpropagation on an RNN.
- Apply the SkipGram Word2Vec technique to implement custom word embeddings.
- Explain the limitations of simple RNNs and how they can be overcome by using long short term memory networks (LSTMs).

Lesson 2

Long Short-Term Memory Networks (LSTMs)

- Understand the functioning of the LSTM via the four LSTM gates: the learning gate, the forget gate, the remember gate, and the use gate.
- Compare architectures such as GRU that can reveal new modeling techniques in combination with the LSTM.

Lesson 3

Implementation of RNN and LSTMs

- Train a simple RNN in PyTorch to do time series prediction.
- Implement a character level sequence RNN.

Lesson 4

Fine Tuning RNN Models

- Fine tune RNN models using hyperparameters.
- Apply key hyperparameters such as learning rate, minibatch size, number of epochs, and number of layers.
- Identify possible starting values and intuitions for the hyperparameters used in RNNs.

Lesson 5

Seq2Seq Architecture

- Implement the components of a Seq2Seq architecture to produce a sequence of words in response to input prompts.
- Implement the key components of a Seq2Seq architecture and understand the way they interact.

Lesson 6

The Limitations of RNNs

- Use recent architectures such as Transformers and BERT to address the limitations of RNNs at solving NLP problems.
- Identify the changes in architecture that occurred during the transition from recurrent networks to Transformer networks.
- Use new Transformer architectures like BERT and GPT3.

Course 4

Building Generative Adversarial Networks

In this course, you'll become familiar with generative adversarial networks (GANs). You will learn how to build and train different GANs architectures to generate new images. Discover, build, and train architectures such as DCGAN, CycleGAN, ProGAN and StyleGAN on diverse datasets including the MNIST dataset, Summer2Winter Yosemite dataset, or CelebA dataset.

Course Project

Face Generation

In this project, you will build and train a custom GAN architecture on the CelebA dataset, leveraging the different skills learned during the course. In particular, you will have to build a custom GAN architecture, including generator and discriminator. You will also experiment with the different loss functions discovered during the course, such as the Binary Cross Entropy loss or the Wasserstein loss. Finally, utilize some of the methods learned to stabilize training, such as label smoothing.

Lesson 1

Generative Adversarial Networks

- Build generator and discriminator using fully connected layers.
- Implement loss functions.
- Train a custom GAN on the MNIST dataset.

Lesson 2

Training a Deep Convolutional GANs

- Build generator and discriminator using convolutional, batch normalization, and fully connected layers.
- Train a DCGAN model on the CIFAR10 dataset.
- Implement evaluation metrics and evaluate generated samples.

Lesson 3

Image to Image Translation

- Implement unpaired data loader.
- Build the CycleGAN generator using residual connection and an encoder-decoder structure.
- Train a CycleGAN model on the summer2winter Yosemite dataset.

Lesson 4

Modern GANs

- Implement Wasserstein loss and gradient penalties.
- Build the ProGAN generator.
- Implement StyleGAN components (adaptive instance normalization).

Meet your instructors.

Erick Galinkin

Principal AI Researcher

Erick Galinkin is a hacker and computer scientist, leading research at the intersection of security and artificial intelligence at Rapid7. He has spoken at numerous industry and academic conferences on topics ranging from malware development to game theory in security.

Giacomo Vianello

Principal Data Scientist

Giacomo Vianello is an end-to-end data scientist with a passion for state-of-the-art, but practical technical solutions. An accomplished speaker with more than 20 invited public talks, he works at Cape Analytics where he brings cutting-edge solutions to the insurance and real estate industries.

Nathan Klarer

Head of ML & COO of Datyra

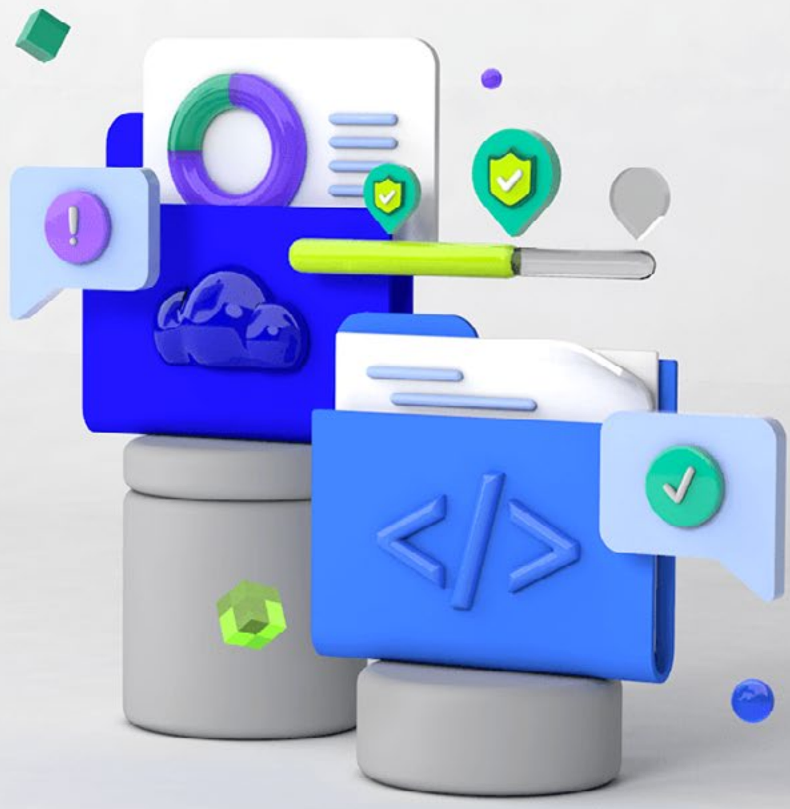
Nathan is a data scientist and entrepreneur. He currently leads a Datyra, a 50-person AI consultancy. He was the first AI team member at $CORZ. Prior to that he founded a VC backed data startup that was acquired. Nathan was named “27 CEO’s Under 27” by Entrepreneur.com and has been featured in *Inc*. and *Forbes*.

Thomas Hossler

Sr Machine Learning Engineer

Thomas is originally a geophysicist but his passion for computer vision led him to become a deep learning engineer at various startups. By creating online courses, he is hoping to make education more accessible. When he is not coding, Thomas can be found in the mountains skiing or climbing.

Udacity’s learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren’t just multiple choice questions or step-by-step guides, but instead require critical thinking.

Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.

Knowledge

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

Custom Study Plans

Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.

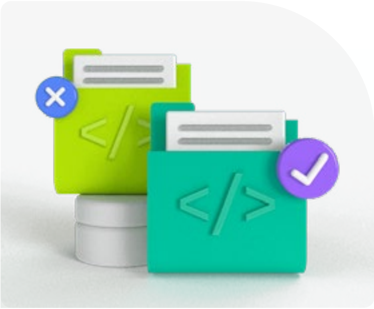
Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.

Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

Our proven approach for building job-ready digital skills.



Experienced Project Reviewers

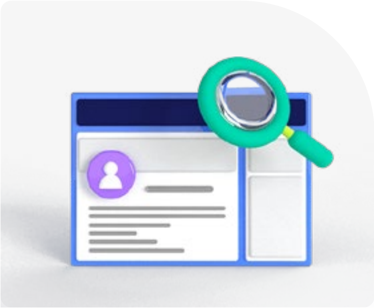
Verify skills mastery.

- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.

Technical Mentor Support

24/7 support unblocks learning.

- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Personal Career Services

Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.

Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.

UDACITY

Learn more at

www.udacity.com/online-learning-for-individuals →