Deep Learning 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

This program will teach you how to become a Deep Learning Engineer, Machine Learning Engineer, AI Engineer, Data Scientist, etc. , Become an expert in neural networks, and learn to implement them in Keras and TensorFlow. Build convolutional networks for image recognition, recurrent networks for sequence generation, generative adversarial networks for image generation, and more. The program is comprised of five projects and accompanying lessons. Each project you build will be an opportunity to demonstrate what you've learned in your lessons. Your completed projects become part of a career portfolio that will demonstrate your mastery of deep learning to potential employers.

Prerequisite Skills

A well-prepared learner is able to:

- Proficient in Python and using NumPy and Pandas
- Proficient in Algebra, Calculus (multivariable derivatives) and linear algebra (matrix multiplication)

Required Hardware

- Webcam
- Microphone
- 64-bit computer

Required Software

- Python 2.7/3.6
- NumPy 1.11
- Anaconda 5.0.1
- Jupyter Notebooks

Version: 3.0.0

Length of Program: 116 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: Introduction to Deep Learning

Introduce yourself to deep learning by applying style transfer to your own images, and gaining experience using development tools such as Anaconda and Jupyter notebooks.

Part 2: Neural Networks

Project: Predicting Bike-Sharing Patterns

In this project, you'll build and train your own Neural Network from scratch to predict the number of bikeshare users on a given day. Good luck!

Supporting Lessons

Lesson	Summary
Introduction to Neural Networks	In this lesson, Luis will teach you the foundations of deep learning and neural networks. You'll also implement gradient descent and backpropagation in python, right here in the classroom!
Implementing Gradient Decent	Mat will introduce you to a different error function and guide you through implementing gradient descent using NumPy matrix multiplication.
Training Neural Networks	Now that you know what neural networks are, in this lesson you will learn several techniques to improve their training.
GPU Workspaces Demo	See a demonstration of GPU workspaces in the Udacity classroom.

Part 3: Convolutional Neural Networks

Project: Dog-Breed Classifier

In this project, you will learn how to build a pipeline to process real-world, user-supplied images. Given an image of a dog, your algorithm will identify an estimate of the canine's breed.

Supporting Lessons

Lesson	Summary
Convolutional Neural Networks	Convolutional Neural Networks allow for spatial pattern recognition. Alexis and Cezanne go over how they help us dramatically improve performance in image classification.
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.
Transfer Learning	Learn how to apply a pre-trained network to a new problem with transfer learning.
Weight Initialization	In this lesson, you'll learn how to find good initial weights for a neural network. Having good initial weights can place the neural network closer to the optimal solution.
Autoencoders	Autoencoders are neural networks used for data compression, image de-noising, and dimensionality reduction. Here, you'll build autoencoders using PyTorch.
Style Transfer	Learn how to use a pre-trained network to extract content and style features from an image. Implement style transfer with your own images!

Part 4: 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.

Project: Generate TV Scripts

Generate a TV script by defining and training a recurrent neural network.

Supporting Lessons

Lesson	Summary
Recurrent Neural Networks	Explore how memory can be incorporated into a deep learning model using recurrent neural networks (RNNs). Learn how RNNs can learn from and generate ordered sequences of data.
Long Short-Term Memory Networks (LSTMs)	Luis explains Long Short-Term Memory Networks (LSTM), and similar architectures which have the benefits of preserving long term memory.
Implementation of RNN and LSTM	Learn how to represent memory in code. Then define and train RNNs in PyTorch and apply them to tasks that involve sequential data.
Hyperparameters	Learn about a number of different hyperparameters that are used in defining and training deep learning models. We'll discuss starting values and intuitions for tuning each hyperparameter.
Embeddings & Word2Vec	In this lesson, you'll learn about embeddings in neural networks by implementing the Word2Vec model.
Sentiment Prediction RNN	Implement a sentiment prediction RNN for predicting whether a movie review is positive or negative!

Part 5: Generative Adversarial Networks

Project: Generate Faces

Define two adversarial networks, a generator and discriminator, and train them until you can generate realistic faces.

Supporting Lessons

Lesson	Summary
Generative Adversarial Networks	Ian Goodfellow, the inventor of GANs, introduces you to these exciting models. You'll also implement your own GAN on the MNIST dataset.
Deep Convolutional GANs	In this lesson you'll implement a Deep Convolution GAN to generate complex color images of house numbers.
Pix2Pix & CycleGAN	Jun-Yan Zhu, one of the creators of the CycleGAN, will lead you through Pix2Pix and CycleGAN formulations that learn to do image-to-image translation tasks!
Implementing a CycleGAN	Cezanne will show you how to implement a CycleGAN in PyTorch and translate images from the summer to winter domains.

Part 6: Deploying a Model

Project: Deploying a Sentiment Analysis Model

In this project, you will build and deploy a neural network which predicts the sentiment of a user-provided movie review. In addition, you will create a simple web app that uses your deployed model.

Supporting Lessons

Lesson	Summary
Introduction to Deployment	This lesson will familiarizing the student with cloud and deployment terminology along with demonstrating how deployment fits within the machine learning workflow.
Building a Model using SageMaker	Learn how to use Amazon's SageMaker service to predict Boston housing prices using SageMaker's built-in XGBoost algorithm.
Deploying and Using a Model	In this lesson students will learn how to deploy a model using SageMaker and how to make use of their deployed model with a simple web application.
Hyperparameter Tuning	In this lesson students will see how to use SageMaker's automatic hyperparameter tuning tools on the Boston housing prices model from lesson 2 and with a sentiment analysis model.

Udacity

Generated Mon Oct 19 23:59:35 PDT 2020