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: 2.0.0

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: Introductions

Part 2: Neural Networks

Project: Your first neural network

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 give you solid foundations on deep learning and neural networks. You'll also implement gradient descent and backpropagation in python right here in the classroom.
Implementing Gradient Descent	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	Introduce & demonstrate the functionality of GPU workspaces in the Udacity classroom

Part 3: Convolutional 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
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.
Convolutional Neural Networks	Alexis explains the theory behind Convolutional Neural Networks and how they help us dramatically improve performance in image classification.
CNNs in TensorFlow	
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 close to the optimal solution. This allows the neural netwo
Autoencoders	Autoencoders are neural networks used for data compression, image denoising, and dimensionality reduction. Here, you'll build autoencoders using TensorFlow.
Transfer Learning in TensorFlow	In practice, most people don't train their own networks on massive datasets. In this lesson, you'll learn how to use a pretrained network on a new problem with transfer learning.

Part 4: Recurrent Networks

Project: Generate TV Scripts

Generate a TV script using a recurrent neural network.

Supporting Lessons

Lesson	Summary
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.
Long Short-Term Memory Networks (LSTM)	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	
Hyperparameters	In this lesson, we'll look at number of different hyperparameter that are important for our deep learning work. We'll discuss starting values and intuitions for tuning each hyperparameter.
Embeddings and 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

Part 5: Generative Adversarial Networks

Project: Generate Faces

Compete two neural networks against each other to 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.

Part 6: Deep Reinforcement Learning

Project: Teach a Quadcopter How to Fly

Build a quadcopter flying agent that learns to take off, hover and land using reinforcement learning.

Supporting Lessons

Lesson	Summary
Introduction to RL	Reinforcement learning is a type of machine learning where the machine or software agent learns how to maximize its performance at a task.
The RL Framework: The Problem	Learn how to mathematically formulate tasks as Markov Decision Processes.
The RL Framework: The Solution	In reinforcement learning, agents learn to prioritize different decisions based on the rewards and punishments associated with different outcomes.
Dynamic Programming	The dynamic programming setting is a useful first step towards tackling the reinforcement learning problem.
Monte Carlo Methods	Write your own implementation of Monte Carlo control to teach an agent to play Blackjack!
Temporal-Difference Methods	Learn about how to apply temporal-difference methods such as Sarsa, Q-Learning, and Expected Sarsa to solve both episodic and continuous tasks.
Solve OpenAI Gym's Taxi-v2 Task	With reinforcement learning now in your toolbox, you're ready to explore a mini project using OpenAI Gym!
RL in Continuous Spaces	Review the fundamental concepts of reinforcement learning, and learn how to adapt traditional algorithms to work with continuous spaces.
Deep Q-Learning	Extend value-based reinforcement learning methods to complex problems using deep neural networks.
Policy-Based Methods	Policy-based methods try to directly optimize for the optimal policy. Learn how they work, and why they are important, especially for domains with continuous action spaces.
Actor-Critic Methods	Learn how to combine value-based and policy-based methods, bringing together the best of both worlds, to solve challenging reinforcement learning problems.