Machine Learning Nanodegree Syllabus

Artificial Neural Networks, TensorFlow, and Machine Learning Algorithms

---

Before You Start

Prerequisites: In order to succeed in this program, we recommend having experience programing in Python, knowledge of inferential statistics, probability, linear algebra and calculus. If you've never programmed before, or want a refresher, you can prepare for this Nanodegree with Lessons 1-4 of Intro to Computer Science.

Educational Objectives: This program will teach you how to become a Machine Learning Engineer, build Machine Learning models and apply them to data sets in fields like finance, healthcare, education, and more.

Length of Program*: ___ Hours
Frequency of Classes: Self-paced
Textbooks required: None
Instructional Tools Available: Video lectures, 1:1 appointments, forum support, In-classroom mentorship

*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.

Project 0: Titanic Survival Exploration

In this project, you will create decision functions that attempt to predict survival outcomes from the 1912 Titanic disaster based on each passenger's features, such as sex and age. You will start with a simple algorithm and increase its complexity until you are able to accurately predict the outcomes for at least 80% of the passengers in the provided data. By the end of this project you'll be able to:

- ➔ Use basic Python code to clean a dataset for analysis
- ➔ Run code to create visualizations from the wrangled data
- ➔ Analyze trends shown in the visualizations and report your conclusions
- ➔ Determine if this program is a good fit for your time and talents

Project 1: Predicting Boston Housing Prices

The Boston housing market is highly competitive, and you want to be the best real estate agent in the area. To compete with your peers, you decide to leverage a few basic machine learning concepts to assist you and a client with finding the best selling price for their home. Luckily, you've come across the Boston Housing dataset which contains aggregated data on various features for houses in Greater Boston communities, including the median value of homes for each of those areas. Your task is to build an optimal model based on a statistical analysis with the tools available. This model will then be used to estimate the best selling price for your clients' homes.

Supporting Lesson Content: Model Evaluation and Validation

Lesson Title	Learning Outcomes
STATISTICAL ANALYSIS	➔ Identify key features of datasets, such as average, mean, median, mode, standard deviation, and quantiles.
DATA MODELING	➔ Learn the basic types of data. ➔ Learn how to handle datasets in sklearn.
EVALUATION AND VALIDATION	➔ Test a model, and use metrics such as accuracy and recall to compare and improve models.
MANAGING ERROR AND COMPLEXITY	➔ Learn the types of error such as overfitting and underfitting. ➔ Learn to identify them using learning curves and model complexity. ➔ Apply techniques such as cross validation to improve your models.

Project 2: Find Donors for CharityML

CharityML is a fictitious charity organization located in the heart of Silicon Valley that was established to provide financial support for people eager to learn machine learning. After nearly 32,000 letters sent to people in the community, CharityML determined that every donation they received came from someone that was making more than $50,000 annually. To expand their potential donor base, CharityML has decided to send letters to residents of California, but to only those most likely to donate to the charity. With nearly *15 million* working Californians, CharityML has brought you on board to help build an algorithm to best identify potential donors and reduce overhead cost of sending mail. Your goal will be evaluate and optimize several different supervised learners to determine which algorithm will provide the highest donation yield while also reducing the total number of letters being sent.

Supporting Lesson Content: Supervised Learning

Lesson Title	Learning Outcomes
SUPERVISED LEARNING TASKS	➔ Difference between Regression and Classification ➔ Learn to predict values with Linear Regression ➔ Learn to predict states using Logistic Regression
DECISION TREES	➔ Train Decision Trees to predict states ➔ Use Entropy to build decision trees recursively
ARTIFICIAL NEURAL NETWORKS	➔ Learn the definition of a Neural Network ➔ Learn to train them using backpropagation ➔ Build a neural network starting from a single perceptron
SUPPORT VECTOR MACHINES	➔ Learn to train a Support Vector Machine to separate data linearly ➔ Use Kernel Methods in order to train SVMs on data that is not linearly separable
NONPARAMETRIC MODELS	➔ Instance Based Learning
BAYESIAN METHODS	➔ Learn the Bayes rule, and how to apply it to predicting data using the Naive Bayes algorithm ➔ Train models using Bayesian Learning ➔ Use Bayesian Inference to create Bayesian Networks of several variables ➔ Bayes NLP Mini-Project
ENSEMBLE OF LEARNERS	➔ Enhance traditional algorithms via boosting ➔ Random forests ➔ AdaBoost

Project 3: Creating Customer Segments

In this project you will apply unsupervised learning techniques on product spending data collected for customers of a wholesale distributor in Lisbon, Portugal to identify customer segments hidden in the data. You will first explore the data by selecting a small subset to sample and determine if any product categories highly correlate with one another. Afterwards, you will preprocess the data by scaling each product category and then identifying (and removing) unwanted outliers. With the good, clean customer spending data, you will apply PCA transformations to the data and implement clustering

algorithms to segment the transformed customer data. Finally, you will compare the segmentation found with an additional labeling and consider ways this information could assist the wholesale distributor with future service changes.

Supporting Lesson Content: Unsupervised Learning

Lesson Title	Learning Outcomes
CLUSTERING	➔ Learn the basics of clustering Data ➔ Cluster data with the K-means algorithm ➔ Cluster data with Single Linkage Clustering ➔ Gaussian models and Expectation Maximization
FEATURE ENGINEERING	➔ Learn to scale features in your data ➔ Learn to select the best features for training data
DIMENSIONALITY REDUCTION	➔ Reduce the dimensionality of the data using Principal Component Analysis and Independent Component Analysis

Project 4: Train a Smartcab to Drive

In the not-so-distant future, taxicab companies across the United States no longer employ human drivers to operate their fleet of vehicles. Instead, the taxicabs are operated by self-driving agents, known as *smartcabs*, to transport people from one location to another within the cities those companies operate. In major metropolitan areas, such as Chicago, New York City, and San Francisco, an increasing number of people have come to depend on *smartcabs* to get to where they need to go as safely and reliably as possible. Although *smartcabs* have become the transport of choice, concerns have arose that a self-driving agent might not be as safe or reliable as human drivers, particularly when considering city traffic lights and other vehicles. To alleviate these concerns, your task as an employee for a national taxicab company is to use reinforcement learning techniques to construct a demonstration of a *smartcab* operating in real-time to prove that both safety and reliability can be achieved.

Supporting Lesson Content: Reinforcement Learning

Lesson Title	Learning Outcomes
REINFORCEMENT LEARNING	➔ Learn the basics of Markov Decision Processes ➔ Find optimal policies using Q-Learning.
GAME THEORY	➔ Poker strategies ➔ Equilibriums ➔ Minimax Strategies

Project 5: Dog Breed Classifier

Supporting Lesson Content: Deep Learning

Lesson Title	Learning Outcomes
MACHINE LEARNING TO DEEP LEARNING	➔ The basics of deep learning, including softmax, one-hot encoding, and cross entropy. ➔ Basic linear classification models such as Logistic Regression, and their associated error function.
DEEP NEURAL NETWORKS	➔ Review: What is a Neural Network? ➔ Activation functions, sigmoid, tanh, and ReLus. ➔ How to train a neural network using backpropagation and the chain rule. ➔ How to improve a neural network using techniques such as regularization and dropout.
CONVOLUTIONAL NEURAL NETWORKS	➔ What is a Convolutional Neural Network? ➔ How CNNs are used in Image recognition.

Project 6: Capstone Proposal

In this capstone project proposal, prior to completing the following Capstone Project, you you will leverage what you've learned throughout the Nanodegree program to author a proposal for solving a problem of your choice by applying machine learning algorithms and techniques. A project proposal encompasses seven key points:

- The project's domain background — the field of research where the project is derived;
- A problem statement — a problem being investigated for which a solution will be defined;
- The datasets and inputs — data or inputs being used for the problem;
- A solution statement — a the solution proposed for the problem given;
- A benchmark model — some simple or historical model or result to compare the defined solution to;
- A set of evaluation metrics — functional representations for how the solution can be measured;
- An outline of the project design — how the solution will be developed and results obtained.

Project 7: Capstone Project

In this capstone project, you will leverage what you've learned throughout the Nanodegree program to solve a problem of your choice by applying machine learning algorithms and techniques. You will first define the problem you want to solve and investigate potential solutions and performance metrics. Next, you will analyze the problem through visualizations and data exploration to have a better understanding of what algorithms and features are appropriate for solving it.
You will then implement your algorithms and metrics of choice, documenting the preprocessing, refinement, and postprocessing steps along the way. Afterwards, you will collect results about the performance of the models used, visualize significant quantities, and validate/justify these values. Finally, you will construct conclusions about your results, and discuss whether your implementation adequately solves the problem.