Chapter – 0 (coursera ML)

**Introduction to**

**Neural Network & Deep Learning**

Course Phases

**1.1 About the Course**

* The *Machine Learning Specialization* is a foundational program. This beginner-friendly program will teach you
* The fundamentals of machine learning and
* How to use these techniques to build real-world AI applications.
* Objectives:
* Master key concepts and gained the practical know-how to:
* Quickly and powerfully apply machine learning to challenging real-world problems.
* It provides a broad introduction to modern machine learning, including:

1. Supervised learning:

* Multiple linear regression,
* Logistic regression,
* Neural networks, and
* Decision trees

1. Unsupervised learning

* Clustering,
* Dimensionality reduction,
* Recommender systems

1. Machine learning innovation

* Evaluating and tuning models,
* Taking a data-centric approach to improving performance, and more.
* Applied Learning Project: By the end of this Specialization, you will be ready to:

1. Build Machine Learning Models in Python

* using popular machine learning libraries NumPy and scikit-learn.

1. Build and train Supervised Machine Learning models for

* prediction and binary classification tasks, including linear regression and logistic regression.

1. Build and train a neural network with TensorFlow to

* perform multi-class classification.

1. Apply best practices for machine learning development

* so that your models generalize to data and tasks in the real world.

1. Build and use

* Decision Trees and
* Tree Ensemble Methods, including:
* random forests and
* boosted trees.

1. Use Unsupervised Learning techniques for unsupervised learning including:

* clustering and
* anomaly detection.

1. Build Recommender Systems with

* a collaborative filtering approach and
* a content-based deep learning method.

1. Build a Deep Reinforcement Learning model.

**PHASES**

Phase A: Machine-Learning

Phase B: Advanced-Learning-Algorithms

Phase C: Unsupervised-Learning, Recommenders, Reinforcement-learning

**1.2 Phase A: Machine-Learning**

* Phase 'A' Objectives:

1. Supervised Machine Learning: Regression and Classification
2. Build machine learning models in Python using popular machine learning libraries NumPy & scikit-learn
3. Build & train supervised machine learning models for 'prediction' & 'binary classification' tasks,

* including linear regression & logistic regression
* Skills you'll gain:
* Linear Regression
* Regularization to Avoid Overfitting
* Logistic Regression for Classification
* Gradient Descent
* Supervised Learning
* Phase modules: There are 3 modules in this course

A1. **Introduction to Machine Learning**

A2. **Regression with multiple input variables**

A3. **Classification**

* **A1. Introduction to Machine Learning**
* objectives:
* Build machine learning models in Python using popular machine learning libraries: 'NumPy' and 'scikit-learn'.
* Build and train supervised machine learning models for prediction and binary classification tasks, including
* ***Linear*** *regression* and
* ***Logistic*** *regression*
* Topics: [20 Topics [146min], 3 quizzes, 2 app items, 4 ungraded labs]

A1.01 Welcome to machine learning! [2]

A1.02 Applications of machine learning [4]

A1.03 What is machine learning? [5]

A1.04 Supervised learning part 1 [6]

A1.05 Supervised learning part 2 [7]

A1.06 Unsupervised learning part 1 [8]

A1.07 Unsupervised learning part 2 [3]

A1.08 Jupyter Notebooks [4]

A1.09 Linear regression model part 1 [10]

A1.10 Linear regression model part 2 [6]

A1.11 Cost function formula [9]

A1.12 Cost function intuition [15]

A1.13 Visualizing the cost function [8]

A1.14 Visualization examples [6]

A1.15 Gradient descent [8]

A1.16 Implementing gradient descent [9]

A1.17 Gradient descent intuition [7]

A1.18 Learning rate [9]

A1.19 Gradient descent for linear regression [6]

A1.20 Running gradient descent [5]

Practice quiz: Supervised vs unsupervised learning [15]

Practice quiz: Regression [10]

Practice quiz: Train the model with gradient descent [10]

Python and Jupyter Notebooks [60]

Optional lab: Model representation [60]

Optional lab: Cost function [60]

Optional lab: Gradient descent [60]

* **A2. Regression with multiple input variables**
* objectives:
* Extend linear regression to handle multiple input features.
* Also learn some methods for improving your model's training and performance, such as:
* Vectorization,
* Feature scaling,
* Feature engineering and
* Polynomial regression.
* Topics: [10 Topics [66min], 2 quizzes, 1 programming assignment, 5 ungraded labs]

A2.01 Multiple features [9]

A2.02 Vectorization part 1 [6]

A2.03 Vectorization part 2 [6]

A2.04 Gradient descent for multiple linear regression [7]

A2.05 Feature scaling part 1 [6]

A2.06 Feature scaling part 2 [7]

A2.07 Checking gradient descent for convergence [5]

A2.08 Choosing the learning rate [6]

A2.09 Feature engineering [3]

A2.10 Polynomial regression [5]

Practice quiz: Multiple linear regression [15]

Practice quiz: Gradient descent in practice [30]

practice lab: Linear regression [180]

Optional lab: Python, NumPy and vectorization [60]

Optional Lab: Multiple linear regression [60]

Optional Lab: Feature scaling and learning rate [60]

Optional lab: Feature engineering and Polynomial regression [60]

Optional lab: Linear regression with scikit-learn [60]

* **A3. Classification**
* objectives:
* Learn how to predict 'categories' using the "logistic regression model".
* You'll learn about the problem of 'overfitting', and
* How to handle overfitting with a method called 'regularization'.
* You'll get to practice implementing logistic regression with regularization at the end.
* Topics: [12 Topics [139min], 2 readings, 4 quizzes, 1 programming assignment, 9 ungraded labs]

A3.01 Motivations [9]

A3.02 Logistic regression [9]

A3.03 Decision boundary [10]

A3.04 Cost function for logistic regression [11]

A3.05 Simplified Cost Function for Logistic Regression [5]

A3.06 Gradient Descent Implementation [6]

A3.07 The problem of overfitting [11]

A3.08 Addressing overfitting [8]

A3.09 Cost function with regularization [9]

A3.10 Regularized linear regression [8]

A3.11 Regularized logistic regression [5]

A3.12 Andrew Ng and Fei-Fei Li on Human-Centered AI [41]

Practice quiz: Classification with logistic regression [30]

Practice quiz: Cost function for logistic regression [30]

Practice quiz: Gradient descent for logistic regression [30]

Practice quiz: The problem of overfitting [30]

practice lab: logistic regression•180 minutes

Optional lab: Classification [60]

Optional lab: Sigmoid function and logistic regression [60]

Optional lab: Decision boundary [60]

Optional lab: Logistic loss [60]

Optional lab: Cost function for logistic regression [60]

Optional lab: Gradient descent for logistic regression [60]

Optional lab: Logistic regression with scikit-learn [60]

Optional lab: Overfitting [60]

Optional lab: Regularization [60]

**1.3 Phase B: Advanced-Learning-Algorithms**

* Phase 'B' Objectives:

1. Build and train a neural network with "TensorFlow" to perform 'multi-class classification'
2. Apply best practices for ML development so that your models

* generalize to data and tasks in the real world

1. Build and use decision trees and tree ensemble methods, including:

* "random forests" and "boosted trees"
* Skills you'll gain:
* Tensorflow
* Advice for Model Development
* Artificial Neural Network
* Xgboost
* Tree Ensembles
* Phase modules: There are 4 modules in this phase

B1. **Neural Networks**

B2. **Neural network training**

B3. **Advice for applying machine learning**

B4. **Decision trees**

* **B1. Neural Networks**
* objectives:
* Learn about 'neural networks' and how to use them for 'classification' tasks.
* Use the TensorFlow framework to build a neural network with just a few lines of code.
* Dive deeper by learning how to code up your own neural network in Python, "from scratch".
* Learn more about how neural network computations are implemented efficiently using parallel processing (vectorization).
* Topics: [17 Topics [139min], 4 quizzes, 1 programming assignment, 1 app item, 3 ungraded labs]

B1.01 Welcome! [2]

B1.02 Neurons and the brain [10]

B1.03 Demand Prediction [16]

B1.04 Example: Recognizing Images [6]

B1.05 Neural network layer [9]

B1.06 More complex neural networks [8]

B1.07 Inference: making predictions (forward propagation) [5]

B1.08 Inference in Code [6]

B1.09 Data in TensorFlow [11]

B1.10 Building a neural network [8]

B1.11 Forward prop in a single layer [5]

B1.12 General implementation of forward propagation [7]

B1.13 Is there a path to AGI? [10]

B1.14 How neural networks are implemented efficiently [4]

B1.15 Matrix multiplication [9]

B1.16 Matrix multiplication rules [9]

B1.17 Matrix multiplication code [6]

Practice quiz: Neural networks intuition [10]

Practice quiz: Neural network model [10]

Practice quiz: TensorFlow implementation [10]

Practice quiz: Neural network implementation in Python [10]

Practice Lab: Neural Networks for Binary Classification [180]

Optional lab: Neurons and Layers [10]

Optional lab: Coffee Roasting in Tensorflow [10]

Optional lab: CoffeeRoastingNumPy [60]

* **B2. Neural network training**
* Objectives:
* How to train your model in TensorFlow
* learn about other important activation functions (besides the sigmoid function), and where to use each type in a neural network.
* How to go beyond binary classification to multiclass classification (3 or more categories).
* Multiclass classification will introduce you to a new 'activation function' and a new 'loss function'.
* Learn about the difference between "multi-CLASS classification" and 'multi-LABEL classification'.
* Learn about the 'Adam optimizer', and why it's an improvement upon 'regular gradient descent' for neural network training.
* Get a brief introduction to other layer types besides the one you've seen so far.
* Topics: [15 Topics [139min], 4 quizzes, 1 programming assignment, 5 ungraded labs]

B2.01 TensorFlow implementation [3]

B2.02 Training Details [13]

B2.03 Alternatives to the sigmoid activation [5]

B2.04 Choosing activation functions [8]

B2.05 Why do we need activation functions? [5]

B2.06 Multiclass [3]

B2.07 Softmax [11]

B2.08 Neural Network with Softmax output [7]

B2.09 Improved implementation of softmax [9]

B2.10 Classification with multiple outputs (Optional) [4]

B2.11 Advanced Optimization [6]

B2.12 Additional Layer Types [8]

B2.13 What is a derivative? (Optional) [22]

B2.14 Computation graph (Optional) [19]

B2.15 Larger neural network example (Optional) [9]

Practice quiz: Neural Network Training [30]

Practice quiz: Activation Functions [30]

Practice quiz: Multiclass Classification [30]

Practice quiz: Additional Neural Network Concepts [30]

Practice Lab: Neural Networks for Multiclass classification [180]

Optional Lab: ReLU activation [60]

Optional Lab: Softmax [60]

Optional Lab: Multiclass [15]

Optional Lab: Derivatives [30]

Optional Lab: Back propagation [30]

* **B3. Advice for applying machine learning**
* Objectives:
* Best practices for 'training' and 'evaluating' your learning algorithms to improve performance.
* It'll cover a wide range of useful advice about the
* machine learning lifecycle,
* tuning your model, and also
* improving your training data.
* Topics: [17 Topics [174min], 3 quizzes, 1 programming assignment, 2 ungraded labs]

B3.01 Deciding what to try next [3]

B3.02 Evaluating a model [10]

B3.03 Model selection and training/cross validation/test sets [13]

B3.04 Diagnosing bias and variance [11]

B3.05 Regularization and bias/variance [10]

B3.06 Establishing a baseline level of performance [9]

B3.07 Learning curves [11]

B3.08 Deciding what to try next revisited [8]

B3.09 Bias/variance and neural networks [10]

B3.10 Iterative loop of ML development [7]

B3.11 Error analysis [8]

B3.12 Adding data [14]

B3.13 Transfer learning: using data from a different task [11]

B3.14 Full cycle of a machine learning project [8]

B3.15 Fairness, bias, and ethics [9]

B3.16 Error metrics for skewed datasets [11]

B3.17 Trading off precision and recall [11]

Practice quiz: Advice for applying machine learning [30]

Practice quiz: Bias and variance [30]

Practice quiz: Machine learning development process [30]

Practice Lab: Advice for Applying Machine Learning [180]

Optional Lab: Model Evaluation and Selection [30]

Optional Lab: Diagnosing Bias and Variance [30]

* **B4. Decision trees**
* Objectives:
* Learn about the decision tree learning algorithm. You'll also
* Learn about variations of the decision tree, including
* random forests and
* boosted trees (XGBoost).
* Topics: [14 Topics [143min], 2 readings, 3 quizzes, 1 programming assignment, 2 ungraded labs]

B4.01 Decision tree model [7]

B4.02 Learning Process [11]

B4.03 Measuring purity [7]

B4.04 Choosing a split: Information Gain [11]

B4.05 Putting it together [9]

B4.06 Using one-hot encoding of categorical features [5]

B4.07 Continuous valued features [6]

B4.08 Regression Trees (optional) [9]

B4.09 Using multiple decision trees [3]

B4.10 Sampling with replacement [3]

B4.11 Random forest algorithm [6]

B4.12 XGBoost [6]

B4.13 When to use decision trees [6]

B4.14 Andrew Ng and Chris Manning on Natural Language Processing [47]

Practice quiz: Decision trees [30]

Practice quiz: Decision tree learning [30]

Practice quiz: Tree ensembles [30]

Practice Lab: Decision Trees [180]

Optional Lab: Decision Trees [30]

Optional Lab: Tree Ensembles [30]

**1.4 Phase C: Unsupervised-Learning, Recommenders,**

**Reinforcement-learning**

* Phase 'C' Objectives:

1. Use unsupervised learning techniques for unsupervised learning including:

* clustering and
* anomaly detection

1. Build recommender systems with

* a collaborative filtering approach and a content-based deep learning method

1. Build a deep reinforcement learning model

* Skills you'll gain:
* Anomaly Detection
* Unsupervised Learning
* Reinforcement Learning
* Collaborative Filtering
* Recommender Systems
* Phase modules: There are 3 modules in this phase

C1. **Unsupervised learning**

C2. **Recommender systems**

C3. **Reinforcement learning**

**C1. Unsupervised learning**

* objectives:
* Learn two key unsupervised learning algorithms:
* clustering and
* anomaly detection
* Topics: [13 Topics [120min], 2 quizzes, 2 programming assignments, 1 app item]

C1.01 Welcome! [3]

C1.02 What is clustering? [4]

C1.03 K-means intuition [6]

C1.04 K-means algorithm [9]

C1.05 Optimization objective [11]

C1.06 Initializing K-means [8]

C1.07 Choosing the number of clusters [7]

C1.08 Finding unusual events [11]

C1.09 Gaussian (normal) distribution [10]

C1.10 Anomaly detection algorithm [11]

C1.11 Developing and evaluating an anomaly detection system [11]

C1.12 Anomaly detection vs. supervised learning [8]

C1.13 Choosing what features to use [14]

Practice quiz: Clustering [30]

Practice quiz: Anomaly detection [30]

Practice Lab: k-means [180]

Practice Lab: Anomaly Detection [180]

**C2. Recommender systems**

* objectives:
* Implementing Recommender systems
* Filtering methods
* Dimensionality/Feature Reduction, PCA
* Topics: [15 Topics [150min], 3 quizzes, 2 programming assignments, 1 ungraded lab]

C2.01 Making recommendations [5]

C2.02 Using per-item features [11]

C2.03 Collaborative filtering algorithm [13]

C2.04 Binary labels: favs, likes and clicks [8]

C2.05 Mean normalization [8]

C2.06 TensorFlow implementation of collaborative filtering [11]

C2.07 Finding related items [6]

C2.08 Collaborative filtering vs Content-based filtering [9]

C2.09 Deep learning for content-based filtering [9]

C2.10 Recommending from a large catalogue [7]

C2.11 Ethical use of recommender systems [10]

C2.12 TensorFlow implementation of content-based filtering [4]

C2.13 Reducing the number of features (optional) [12]

C2.14 PCA algorithm (optional) [17]

C2.15 PCA in code (optional) [11]

quiz: Collaborative Filtering [30]

quiz: Recommender systems implementation [30]

quiz: Content-based filtering [30]

Practice Lab: Collaborative Filtering Recommender Systems [180]

Practice Lab: Deep Learning for Content-Based Filtering [180]

Practice Lab: PCA and data visualization (optional) [30]

**C3. Reinforcement learning**

* objectives:
* Learn about reinforcement learning,
* Build a deep Q-learning neural network in order to land a virtual lunar lander on Mars!
* Topics: [18 Topics [163min], 3 readings, 3 quizzes, 1 programming assignment, 1 ungraded lab]

C3.01 What is Reinforcement Learning? [8]

C3.02 Mars rover example [6]

C3.03 The Return in reinforcement learning [10]

C3.04 Making decisions: Policies in reinforcement learning [2]

C3.05 Review of key concepts [5]

C3.06 State-action value function definition [10]

C3.07 State-action value function example [5]

C3.08 Bellman Equation [12]

C3.09 Random (stochastic) environment (Optional) [8]

C3.10 Example of continuous state space applications [6]

C3.11 Lunar lander [5]

C3.12 Learning the state-value function [16]

C3.13 Algorithm refinement: Improved neural network architecture [3]

C3.14 Algorithm refinement: ϵ-greedy policy [8]

C3.15 Algorithm refinement: Mini-batch and soft updates (optional) [11]

C3.16 The state of reinforcement learning [2]

C3.17 Summary and thank you [3]

C3.18 Andrew Ng and Chelsea Finn on AI and Robotics [33]

quizz: Reinforcement learning introduction [30]

quizz: State-action value function [30]

quizz: Continuous state spaces [30]

Practice Lab: Reinforcement Learning [180]

Practice Lab: State-action value function (optional lab) [60]

**Old Content**

**chapter 01:** Intro

**chapter 02:** Linear Regression (single Variable)

**chapter 03:** Linear Algebra Review

**chapter 04:** Linear Regression (Multiple Variables)

**chapter 05:** Octave Tutorial

**chapter 06:** Logistic Regression

**chapter 07:** Regularization

**chapter 08:** Neural Networks Representation

**chapter 09:** Neural Networks Learning

**chapter 10:** Advice For Applying Machine Learning

**chapter 11:** Machine Learning System Design

**chapter 12:** Support Vector Machines

**chapter 13:** Clustering

**chapter 14:** Dimensionality Reduction

**chapter 15:** Anomaly Detection

**chapter 16:** Recommender Systems

**chapter 17:** Large Scale Machine Learning

**chapter 18:** Application Example

**chapter 19:** Conclusion

**Chapter 01: Intro**

1.1 — What Is Machine Learning

1.2 — Supervised Learning

1.3 — Unsupervised Learning

**Chapter 02: Linear Regression (single Variable)**

2.1 — Model Representation

2.2 — Cost Function — intro

2.3 — Cost Function Intuition p1

2.4 — Cost Function Intuition p2

2.5 — Gradient Descent

2.6 — Gradient Descent Intuition

2.7 — Gradient Descent For Linear Regression

2.8 — What's Next

**Chapter 03: Linear Algebra Review**

3.1 — Matrices And Vectors

3.2 — Addition And Scalar Multiplication

3.3 — Matrix Vector Multiplication

3.4 — Matrix-Matrix Multiplication

3.5 — Matrix Multiplication Properties

3.6 — Inverse And Transpose

**Chapter 04: Linear Regression (Multiple Variables)**

4.1 — Multiple Features

4.2 — Gradient Descent For Multiple Features

4.3 — Gradient Descent In Practice 1

4.4 — Gradient Descent In Practice 2

4.5 — Features And Polynomial Regression

4.6 — Normal Equation

4.7 — Normal Equation Non-invertibility

**Chapter 05: Octave Tutorial**

5.1 — Octave Tutorial \_ Basic Operations

5.2 — Octave Tutorial \_ Moving Data Around

5.3 — Octave Tutorial \_ Computing On Data

5.4 — Octave Tutorial \_ Plotting Data

5.5 — Octave Tutorial \_ While If Statements And Functions

5.6 — Octave Tutorial \_ Vectorization

5.7 — Octave Tutorial \_ Programming Exercises

**Chapter 06: Logistic Regression**

6.1 — Classification

6.2 — Hypothesis Representation

6.3 — Decision Boundary

6.4 — Cost Function

6.5 — Simplified Cost Function And Gradient Descent

6.6 — Advanced Optimization

6.7 — Multiclass Classification 'One Vs All'

**Chapter 07: Regularization**

7.1 — The Problem Of Overfitting

7.2 — Cost Function

7.3 — Regularized Linear Regression

7.4 — Regularized Logistic Regression

**Chapter 08: Neural Networks Representation**

8.1 — Non Linear Hypotheses

8.2 — Neurons And The Brain

8.3 — Model Representation 1

8.4 — Model Representation 2

8.5 — Examples And Intuitions 1

8.6 — Examples And Intuitions 2

8.7 — Multiclass Classification

**Chapter 09: Neural Networks Learning**

9.1 — Cost Function

9.2 — Backpropagation Algorithm

9.3 — Backpropagation Intuition

9.4 — Implementation Note: Unrolling Parameter

9.5 — Gradient Checking

9.6 — Random Initialization

9.7 — Putting It Together

9.8 — Autonomous Driving Example

**Chapter 10: Advice For Applying Machine Learning**

10.1 — Deciding What To Try Next

10.2 — Evaluating a Hypothesis

10.3 — Model Selection And Training/validation/test-sets

10.4 — Diagnosing Bias Vs Variance

10.5 — Regularization And Bias/Variance

10.6 — Learning Curves

10.7 — Deciding What To Do Next (revisited)

**Chapter 11: Machine Learning System Design**

11.1 — Prioritizing What To Work On: spam classification example

11.2 — Error Analysis

11.3 — Error Metrics For Skewed Classes

11.4 — Trading Off Precision And Recall

11.5 — Data For Machine Learning

**Chapter 12: Support Vector Machines**

12.1 — Optimization Objective

12.2 — Large Margin Intuition

12.3 — Mathematics Behind Large Margin Classification

12.4 — Kernels-i

12.5 — Kernels-ii

12.6 — Using An SVM

**Chapter 13: Clustering**

13.1 — Unsupervised Learning Introduction

13.2 — KMeans Algorithm

13.3 — Optimization Objective

13.4 — Random Initialization

13.5 — Choosing The Number Of Clusters

**Chapter 14: Dimensionality Reduction**

14.1 — Motivation I - Data Compression

14.2 — Motivation II - Visualization

14.3 — Principal Component Analysis - Problem formulation

14.4 — Principal Component Analysis Algorithm

14.5 — Choosing The Number Of Principal Components

14.6 — Reconstruction From Compressed Representation

14.7 — Advice For Applying PCA

**Chapter 15: Anomaly Detection**

15.1 — Problem Motivation

15.2 — Gaussian Distribution

15.3 — Algorithm

15.4 — Developing And Evaluating An Anomaly Detection system

15.5 — Anomaly Detection Vs Supervised Learning

15.6 — Choosing What Features To Use

15.7 — Multivariate Gaussian Distribution

15.8 — Anomaly Detection Using The Multivariate Gaussian Distribution

**Chapter 16: Recommender Systems**

16.1 — Problem Formulation

16.2 — Content Based Recommendations

16.3 — Collaborative Filtering

16.4 — Collaborative Filtering Algorithm

16.5 — Vectorization Low Rank Matrix Factorization

16.6 — Implementational Detail Mean Normalization

**Chapter 17: Large Scale Machine Learning**

17.1 — Learning With Large Datasets

17.2 — Stochastic Gradient Descent

17.3 — Mini Batch Gradient Descent

17.4 — Stochastic Gradient Descent Convergence

17.5 — Online Learning

17.6 — Map Reduce And Data Parallelism

**Chapter 18: Application Example**

18.1 — Problem Description And Pipeline

18.2 — Sliding Windows

18.3 — Getting Lots Of Data Artificial Data Synthesis

18.4 — Ceiling Analysis: What Part to Work on next

**Chapter 19**

19 — Conclusion Summary And Thank You