Course Overview

Hello, and welcome to this course!

This course introduces the fundamentals of machine learning with a focus on practical implementation using PyTorch. It covers the basics of tensors, datasets, linear regression, stochastic gradient descent, and logistic regression, setting a solid foundation for understanding and building machine learning algorithms.

This course is one in the IBM AI Engineering Professional Certificate series of courses. We recommend after finishing this course, you take the second part of the course, Deep Learning with PyTorch to cover the rest of the topics on neural networks and Pytorch.

Follow the links below to learn more about each of the AI Engineering Professional Certificate series of courses and see how these programs can benefit you and advance your career.

- Machine Learning with Python
- Introduction to Deep Learning & Neural Networks with Keras
- Deep Learning with Keras and Tensorflow
- Deep Learning with PyTorch
- AI Capstone Project with Deep Learning
- Generative AI and LLMs: Architecture and Data Preparation
- Gen AI Model Foundations for NLP & Language Understanding
- Generative AI Language Modeling with Transformers
- Generative AI Engineering and Fine-Tuning Transformers
- Generative AI Advance Fine-Tuning for LLMs
- Fundamentals of Building AI Agents using RAG and LangChain
- Project: Generative AI with RAG and LangChain

Prerequisites

To get the most out of this course, you should be comfortable with the following topics and technologies:

- Basic knowledge of Python
- Machine learning
- Neural networks

This course requires basic knowledge of Pytorch.

Course Objectives

After completing this course, you will be able to:

- Perform tensor operations in PyTorch
- Implement and train linear regression models from scratch using PyTorch’s functionalities
- Explain the concepts of logistic regression and apply them to classification problems
- Handle data and training models using gradient descent for optimization

Course Outline

This course has six modules, which are listed below. We encourage you to set aside several hours each week to complete all modules in 6 weeks. Consistency will best help you achieve your learning goals!

You will derive the maximum benefit from viewing all videos and readings and solidifying that knowledge by completing all of the activities.

Module 1: Tensor and Data Set

This module provides an overview of tensors and datasets. It will cover the appropriate methods to classify the type of data in a tensor and the type of tensor. You will learn the basics of 1-D and 2-D tensors and the Numel method. Then, you will learn to differentiate between simple and partial derivatives. The module lists the different attributes that PyTorch uses to calculate a derivative. You will build a simple dataset class, object, and a dataset for images.

Module 2: Linear Regression

This module describes linear regression. You will learn about classes and how to build custom modules using nn.Modules to make predictions. Then, you will explore the state_dict() method that returns a Python dictionary. Then, you will learn how to train the model and define a dataset and the noise assumption. You will see how to minimize the cost and calculate loss using PyTorch. You will understand the gradient descent method and how to apply it to the cost function. You will learn to determine the bias and slope using the gradient descent method and define the cost surface.

Module 3: Linear Regression PyTorch Way

This module covers implementing stochastic gradient descent using PyTorch’s data loader. Then, you will explore batch processing techniques for efficient model training. You will compare Mini-Batch Gradient Descent and Stochastic Gradient Descent. Next, you will learn about convergence rate and using PyTorch’s optimization modules. Finally, you will learn the best practices for splitting data to ensure robust model evaluation and how hyperparameters are applied to train data.

Module 4: Multiple Input Output Linear Regression

In this module, you will learn to use the class linear to perform linear regression in multiple dimensions. In addition, you will learn about model parameters and how to calculate cost and perform gradient descent in PyTorch. You will learn to extend linear regression for multiple outputs.

Module 5: Logistic Regression for Classification

In this module, you will learn the fundamentals of linear classifiers and logistic regression. You will learn to use the nn.sequential model to build neural networks in PyTorch. You will implement logistic regression for prediction. The module also covers statistical concepts like Bernoulli distribution and maximum likelihood estimation underpinning logistic regression. In addition, you will understand and implement the cross-entropy loss function.

Module 6: Final Project

In this module, you will implement the final project applying all concepts learned. You will build a logistic regression model to predict the outcomes of League of Legends matches. Leveraging various in-game statistics, this project will utilize your knowledge of PyTorch, logistic regression, and data handling to create a robust predictive model.

Tools/software used

In this course, you will use free versions or trials of several tools, including *Jupyter Notebook.* The course does not require you to pay to complete any of the activities. You may wish to upgrade to a paid version, though you will need to incur those costs.

Congratulations on taking these steps to further your knowledge and career! Enjoy your journey.