Machine Learning

Stanford University

About this Course

Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. Machine learning is so pervasive today that you probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human-level AI. In this class, you will learn about the most effective machine learning techniques, and gain practice implementing them and getting them to work for yourself. More importantly, you'll learn about not only the theoretical underpinnings of learning, but also gain the practical know-how needed to quickly and powerfully apply these techniques to new problems. Finally, you'll learn about some of Silicon Valley's best practices in innovation as it pertains to machine learning and AI.

This course provides a broad introduction to machine learning, datamining, and statistical pattern recognition. Topics include: (i) Supervised learning (parametric/non-parametric algorithms, support vector machines, kernels, neural networks). (ii) Unsupervised learning (clustering, dimensionality reduction, recommender systems, deep learning). (iii) Best practices in machine learning (bias/variance theory; innovation process in machine learning and AI). The course will also draw from numerous case studies and applications, so that you'll also learn how to apply learning algorithms to building smart robots (perception, control), text understanding (web search, anti-spam), computer vision, medical informatics, audio, database mining, and other areas.

^ Show less

Taught by: Andrew Ng, Associate Professor, Stanford University; Chief Scientist, Baidu; Chairman and Co-founder, Coursera

Help Center

Language	English, **Subtitles:** Spanish, Japanese, Chinese (Simplified) Volunteer to translate subtitles for this course

How To Pass	Pass all graded assignments to complete the course.
User Ratings	★★★★⯪ 4.9 stars

Syllabus

WEEK 1

Introduction

Welcome to Machine Learning! In this module, we introduce the core idea of teaching a computer to learn concepts using data—without being explicitly programmed. The Course Wiki is under construction. Please visit the resources tab for the most complete and up-to-date information. Less

4 videos, 10 readings

1. **Video:** Welcome to Machine Learning!
2. **Reading:** Machine Learning Honor Code
3. **Video:** Welcome
4. **Reading:** What is Machine Learning?
5. **Reading:** How to Use Discussion Forums
6. **Video:** Supervised Learning
7. **Reading:** Supervised Learning
8. **Video:** Unsupervised Learning
9. **Reading:** Unsupervised Learning
10. **Reading:** Who are Mentors?
11. **Reading:** Get to Know Your Classmates
12. **Reading:** Frequently Asked Questions
13. **Reading:** Lecture Slides
14. **Reading:** Machine Learning Course Wiki

Graded: Introduction

Help Center

Linear Regression with One Variable

Linear regression predicts a real-valued output based on an input value. We discuss the application of linear regression to housing price prediction, present the notion of a cost function, and introduce the gradient descent method for learning.

7 videos, 8 readings

1. **Video:** Model Representation
2. **Reading:** Model Representation
3. **Video:** Cost Function
4. **Reading:** Cost Function
5. **Video:** Cost Function - Intuition I
6. **Reading:** Cost Function - Intuition I
7. **Video:** Cost Function - Intuition II
8. **Reading:** Cost Function - Intuition II
9. **Video:** Gradient Descent
10. **Reading:** Gradient Descent
11. **Video:** Gradient Descent Intuition
12. **Reading:** Gradient Descent Intuition
13. **Video:** Gradient Descent For Linear Regression
14. **Reading:** Gradient Descent For Linear Regression
15. **Reading:** Lecture Slides

Graded: Linear Regression with One Variable

Linear Algebra Review

This optional module provides a refresher on linear algebra concepts. Basic understanding of linear algebra is necessary for the rest of the course, especially as we begin to cover models with multiple variables.

6 videos, 1 reading, 1 reading

1. **Video:** Matrices and Vectors
2. **Video:** Addition and Scalar Multiplication
3. **Video:** Matrix Vector Multiplication
4. **Video:** Matrix Matrix Multiplication
5. **Video:** Matrix Multiplication Properties
6. **Video:** Inverse and Transpose
7. **Reading:** Lecture Slides

Help Center

8. **Practice Quiz:** Linear Algebra

WEEK 2

Linear Regression with Multiple Variables

What if your input has more than one value? In this module, we show how linear regression can be extended to accommodate multiple input features. We also discuss best practices for implementing linear regression.

8 videos, 16 readings

1. **Reading:** Setting Up Your Programming Assignment Environment
2. **Reading:** Installing MATLAB
3. **Reading:** Installing Octave on Windows
4. **Reading:** Installing Octave on Mac OS X (10.10 Yosemite and 10.9 Mavericks and Later)
5. **Reading:** Installing Octave on Mac OS X (10.8 Mountain Lion and Earlier)
6. **Reading:** Installing Octave on GNU/Linux
7. **Reading:** More Octave/MATLAB resources
8. **Video:** Multiple Features
9. **Reading:** Multiple Features
10. **Video:** Gradient Descent for Multiple Variables
11. **Reading:** Gradient Descent For Multiple Variables
12. **Video:** Gradient Descent in Practice I - Feature Scaling
13. **Reading:** Gradient Descent in Practice I - Feature Scaling
14. **Video:** Gradient Descent in Practice II - Learning Rate
15. **Reading:** Gradient Descent in Practice II - Learning Rate
16. **Video:** Features and Polynomial Regression
17. **Reading:** Features and Polynomial Regression
18. **Video:** Normal Equation
19. **Reading:** Normal Equation
20. **Video:** Normal Equation Noninvertibility
21. **Reading:** Normal Equation Noninvertibility

Help Center

22. **Video:** Working on and Submitting Programming Assignments
23. **Reading:** Programming tips from Mentors
24. **Reading:** Lecture Slides

Graded: Linear Regression with Multiple Variables

Octave/Matlab Tutorial

This course includes programming assignments designed to help you understand how to implement the learning algorithms in practice. To complete the programming assignments, you will need to use Octave or MATLAB. This module introduces Octave/Matlab and shows you how to submit an assignment. Less

6 videos, 1 reading

1. **Video:** Basic Operations
2. **Video:** Moving Data Around
3. **Video:** Computing on Data
4. **Video:** Plotting Data
5. **Video:** Control Statements: for, while, if statement
6. **Video:** Vectorization
7. **Reading:** Lecture Slides
8. **Programming:** Linear Regression

Graded: Octave/Matlab Tutorial

WEEK 3

Logistic Regression

Logistic regression is a method for classifying data into discrete outcomes. For example, we might use logistic regression to classify an email as spam or not spam. In this module, we introduce the notion of classification, the cost function for logistic regression, and the application of logistic regression to multi-class classification. Less

Help Center

7 videos, 8 readings

1. **Video:** Classification
2. **Reading:** Classification

3. **Video:** Hypothesis Representation
4. **Reading:** Hypothesis Representation
5. **Video:** Decision Boundary
6. **Reading:** Decision Boundary
7. **Video:** Cost Function
8. **Reading:** Cost Function
9. **Video:** Simplified Cost Function and Gradient Descent
10. **Reading:** Simplified Cost Function and Gradient Descent
11. **Video:** Advanced Optimization
12. **Reading:** Advanced Optimization
13. **Video:** Multiclass Classification: One-vs-all
14. **Reading:** Multiclass Classification: One-vs-all
15. **Reading:** Lecture Slides

Graded: Logistic Regression

Regularization

Machine learning models need to generalize well to new examples that the model has not seen in practice. In this module, we introduce regularization, which helps prevent models from overfitting the training data.

4 videos, 5 readings

1. **Video:** The Problem of Overfitting
2. **Reading:** The Problem of Overfitting
3. **Video:** Cost Function
4. **Reading:** Cost Function
5. **Video:** Regularized Linear Regression
6. **Reading:** Regularized Linear Regression
7. **Video:** Regularized Logistic Regression
8. **Reading:** Regularized Logistic Regression
9. **Reading:** Lecture Slides
10. **Programming:** Logistic Regression

Graded: Regularization

Help Center

WEEK 4

Neural Networks: Representation

Neural networks is a model inspired by how the brain works. It is widely used today in many applications: when your phone interprets and understand your voice commands, it is likely that a neural network is helping to understand your speech; when you cash a check, the machines that automatically read the digits also use neural networks. Less

7 videos, 6 readings

1. **Video:** Non-linear Hypotheses
2. **Video:** Neurons and the Brain
3. **Video:** Model Representation I
4. **Reading:** Model Representation I
5. **Video:** Model Representation II
6. **Reading:** Model Representation II
7. **Video:** Examples and Intuitions I
8. **Reading:** Examples and Intuitions I
9. **Video:** Examples and Intuitions II
10. **Reading:** Examples and Intuitions II
11. **Video:** Multiclass Classification
12. **Reading:** Multiclass Classification
13. **Reading:** Lecture Slides
14. **Programming:** Multi-class Classification and Neural Networks

Graded: Neural Networks: Representation

WEEK 5

Help Center

Neural Networks: Learning

In this module, we introduce the backpropagation algorithm that is used to help learn parameters for a neural network. At the end of this module, you will be implementing your own neural network for digit recognition.

8 videos, 8 readings

1. **Video:** Cost Function
2. **Reading:** Cost Function
3. **Video:** Backpropagation Algorithm
4. **Reading:** Backpropagation Algorithm
5. **Video:** Backpropagation Intuition
6. **Reading:** Backpropagation Intuition
7. **Video:** Implementation Note: Unrolling Parameters
8. **Reading:** Implementation Note: Unrolling Parameters
9. **Video:** Gradient Checking
10. **Reading:** Gradient Checking
11. **Video:** Random Initialization
12. **Reading:** Random Initialization
13. **Video:** Putting It Together
14. **Reading:** Putting It Together
15. **Video:** Autonomous Driving
16. **Reading:** Lecture Slides
17. **Programming:** Neural Network Learning

Graded: Neural Networks: Learning

WEEK 6

Advice for Applying Machine Learning

Applying machine learning in practice is not always straightforward. In this module, we share best practices for applying machine learning in practice, and discuss the best ways to evaluate performance of the learned models.

7 videos, 2 readings

1. **Video:** Deciding What to Try Next
2. **Video:** Evaluating a Hypothesis
3. **Reading:** Evaluating a Hypothesis
4. **Video:** Model Selection and Train/Validation/Test Sets

Help Center

5. **Video:** Diagnosing Bias vs. Variance
6. **Video:** Regularization and Bias/Variance
7. **Video:** Learning Curves
8. **Video:** Deciding What to Do Next Revisited
9. **Reading:** Lecture Slides
10. **Programming:** Regularized Linear Regression and Bias/Variance

Graded: Advice for Applying Machine Learning

Machine Learning System Design

To optimize a machine learning algorithm, you'll need to first understand where the biggest improvements can be made. In this module, we discuss how to understand the performance of a machine learning system with multiple parts, and also how to deal with skewed data. Less

5 videos, 1 reading

1. **Video:** Prioritizing What to Work On
2. **Video:** Error Analysis
3. **Video:** Error Metrics for Skewed Classes
4. **Video:** Trading Off Precision and Recall
5. **Video:** Data For Machine Learning
6. **Reading:** Lecture Slides

Graded: Machine Learning System Design

WEEK 7

Support Vector Machines

Support vector machines, or SVMs, is a machine learning algorithm for classification. We introduce the idea and intuitions behind SVMs and discuss how to use it in practice.

6 videos, 1 reading

1. **Video:** Optimization Objective
2. **Video:** Large Margin Intuition
3. **Video:** Mathematics Behind Large Margin Classification

Help Center

4. **Video:** Kernels I
5. **Video:** Kernels II
6. **Video:** Using An SVM
7. **Reading:** Lecture Slides
8. **Programming:** Support Vector Machines

Graded: Support Vector Machines

WEEK 8

Unsupervised Learning

We use unsupervised learning to build models that help us understand our data better. We discuss the k-Means algorithm for clustering that enable us to learn groupings of unlabeled data points.

5 videos, 1 reading

1. **Video:** Unsupervised Learning: Introduction
2. **Video:** K-Means Algorithm
3. **Video:** Optimization Objective
4. **Video:** Random Initialization
5. **Video:** Choosing the Number of Clusters
6. **Reading:** Lecture Slides

Graded: Unsupervised Learning

Dimensionality Reduction

In this module, we introduce Principal Components Analysis, and show how it can be used for data compression to speed up learning algorithms as well as for visualizations of complex datasets.

7 videos, 1 reading

1. **Video:** Motivation I: Data Compression
2. **Video:** Motivation II: Visualization
3. **Video:** Principal Component Analysis Problem Formulation
4. **Video:** Principal Component Analysis Algorithm

Help Center

5. **Video:** Reconstruction from Compressed Representation
6. **Video:** Choosing the Number of Principal Components
7. **Video:** Advice for Applying PCA
8. **Reading:** Lecture Slides
9. **Programming:** K-Means Clustering and PCA

Graded: Principal Component Analysis

WEEK 9

Anomaly Detection

Given a large number of data points, we may sometimes want to figure out which ones vary significantly from the average. For example, in manufacturing, we may want to detect defects or anomalies. We show how a dataset can be modeled using a Gaussian distribution, and how the model can be used for anomaly detection. Less

8 videos, 1 reading

1. **Video:** Problem Motivation
2. **Video:** Gaussian Distribution
3. **Video:** Algorithm
4. **Video:** Developing and Evaluating an Anomaly Detection System
5. **Video:** Anomaly Detection vs. Supervised Learning
6. **Video:** Choosing What Features to Use
7. **Video:** Multivariate Gaussian Distribution
8. **Video:** Anomaly Detection using the Multivariate Gaussian Distribution
9. **Reading:** Lecture Slides

Graded: Anomaly Detection

Help Center

Recommender Systems

When you buy a product online, most websites automatically recommend other products that you may like. Recommender systems look at patterns of activities between different users and different products to produce these recommendations. In this module, we introduce recommender algorithms such as the collaborative filtering algorithm and low-rank matrix factorization. Less

6 videos, 1 reading

1. **Video:** Problem Formulation
2. **Video:** Content Based Recommendations
3. **Video:** Collaborative Filtering
4. **Video:** Collaborative Filtering Algorithm
5. **Video:** Vectorization: Low Rank Matrix Factorization
6. **Video:** Implementational Detail: Mean Normalization
7. **Reading:** Lecture Slides
8. **Programming:** Anomaly Detection and Recommender Systems

Graded: Recommender Systems

WEEK 10

Large Scale Machine Learning

Machine learning works best when there is an abundance of data to leverage for training. In this module, we discuss how to apply the machine learning algorithms with large datasets.

6 videos, 1 reading

1. **Video:** Learning With Large Datasets
2. **Video:** Stochastic Gradient Descent
3. **Video:** Mini-Batch Gradient Descent
4. **Video:** Stochastic Gradient Descent Convergence
5. **Video:** Online Learning
6. **Video:** Map Reduce and Data Parallelism
7. **Reading:** Lecture Slides

Graded: Large Scale Machine Learning

Help Center

WEEK 11

Application Example: Photo OCR

Identifying and recognizing objects, words, and digits in an image is a challenging task. We discuss how a pipeline can be built to tackle this problem and how to analyze and improve the performance of such a system.

5 videos, 1 reading

1. **Video:** Problem Description and Pipeline
2. **Video:** Sliding Windows
3. **Video:** Getting Lots of Data and Artificial Data
4. **Video:** Ceiling Analysis: What Part of the Pipeline to Work on Next
5. **Reading:** Lecture Slides
6. **Video:** Summary and Thank You

Graded: Application: Photo OCR

View Less

How It Works

GENERAL

1. How do I pass the course?

To earn your Course Certificate, you'll need to earn a passing grade on each of the required assignments—these can be quizzes, peer-graded assignments, or programming assignments. Videos, readings, and practice exercises are there to help you prepare for the graded assignments.

2. What do start dates and end dates mean?

Most courses have sessions that run multiple times a year — each with a specific start and end date. Once you enroll for a Certificate, you'll have access to all videos, readings, quizzes, and programming assignments (if applicable). Peer-graded assignments can only be submitted and reviewed once your session has begun. If you choose to explore the course without purchasing, you may not be able to access certain assignments. If you don't finish all graded assignments before the end of the session, you can enroll in the next session. Your progress will be saved and you'll be able to pick up where you left off when the next session begins.

Help Center

3. What are due dates? Is there a penalty for submitting my work after a due date?

Within each session there are suggested due dates to help you manage your schedule and keep coursework from piling up. Quizzes and programming assignments can be submitted late without consequence. However, it is possible that you won't receive a grade if you submit your peer-graded assignment too late because classmates usually review assignment within three days of the assignment deadline.

4. Can I re-attempt an assignment?

Yes. If you want to improve your grade, you can always try again. If you're re-attempting a peer-graded assignment, re-submit your work as soon as you can to make sure there's enough time for your classmates to review your work. In some cases you may need to wait before re-submitting a programming assignment or quiz. We encourage you to review course material during this delay.

⌃ Show less

PROGRAMMING ASSIGNMENTS

Programming assignments require you to write and run a computer program to solve a problem.

1. What are programming assignments?

Programming assignments include both assignment instructions and assignment parts. Instructions may include a link to a downloadable starter package that includes starter code, detailed guidelines, and other resources. Assignment parts are similar to individual quiz questions. Each part is a single coding task that can be completed one at a time.

2. How are programming assignments graded?

Programming assignments are graded automatically. If they use a built-in-algorithm you'll see your grade within seconds. If they use a custom grader, you may need to wait up to an hour.

3. Can I resubmit a programming assignment?

You can resubmit all programming assignments to improve your grade. Follow the same steps as submitting a new assignment.

Help Center

4. What do I do if I have trouble submitting my assignment?

If you have trouble submitting your assignment, we encourage you to visit your course Discussion Forums as many of your peers are likely to have had similar problems and have found a solution. Each programming assignment has its own sub-forum to discuss with peers.

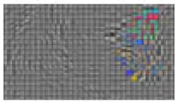
⌃ Show less

View the course in catalog

Related Courses

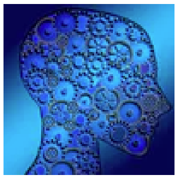
Machine Learning Foundations: A Case Study Approach

University of Washington



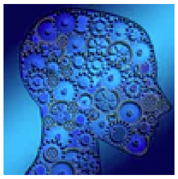
Neural Networks for Machine Learning

University of Toronto



Machine Learning: Regression

University of Washington



Machine Learning: Classification

University of Washington

Machine Learning: Clustering & Retrieval

University of Washington

Help Center