
Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization

deeplearning.ai

About this Course

This course will teach you the "magic" of getting deep learning to work well. Rather than the deep learning process being a black box, you will understand what drives performance, and be able to more systematically get good results. You will also learn TensorFlow.

After 3 weeks, you will:

- Understand industry best-practices for building deep learning applications.
- Be able to effectively use the common neural network "tricks", including initialization, L2 and dropout regularization, Batch normalization, gradient checking,
- Be able to implement and apply a variety of optimization algorithms, such as mini-batch gradient descent, Momentum, RMSprop and Adam, and check for their convergence.
- Understand new best-practices for the deep learning era of how to set up train/dev/test sets and analyze bias/variance
- Be able to implement a neural network in TensorFlow.

This is the second course of the Deep Learning Specialization.

^ Show less



Taught by: Andrew Ng, Co-founder, Coursera; Adjunct Professor, Stanford University; formerly head of Baidu AI Group/Google Brain



Taught by: Head Teaching Assistant - Kian Katanforoosh, Adjunct Lecturer at Stanford University, deeplearning.ai, Ecole Centrale Paris



Taught by: Teaching Assistant - Younes Bensouda Mourri, Mathematical & Computational Sciences, Stanford University, deeplearning.ai

Basic Info	Course 2 of 5 in the Deep Learning Specialization
Level	Beginner
Commitment	3 weeks, 3-6 hours per week
Language	English, Subtitles: Chinese (Traditional), 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

Practical aspects of Deep Learning

📄 15 videos

1. **Video:** Train / Dev / Test sets
2. **Video:** Bias / Variance
3. **Video:** Basic Recipe for Machine Learning
4. **Video:** Regularization
5. **Video:** Why regularization reduces overfitting?
6. **Video:** Dropout Regularization
7. **Video:** Understanding Dropout
8. **Video:** Other regularization methods
9. **Video:** Normalizing inputs
10. **Video:** Vanishing / Exploding gradients
11. **Video:** Weight Initialization for Deep Networks
12. **Video:** Numerical approximation of gradients
13. **Video:** Gradient checking
14. **Video:** Gradient Checking Implementation Notes
15. **Notebook:** Initialization
16. **Notebook:** Regularization
17. **Notebook:** Gradient Checking
18. **Video:** Yoshua Bengio interview

Show less

 **Graded:** Practical aspects of deep learning

 **Graded:** Initialization

 **Graded:** Regularization

 **Graded:** Gradient Checking


WEEK 2


Optimization algorithms

 11 videos

1. **Video:** Mini-batch gradient descent
2. **Video:** Understanding mini-batch gradient descent
3. **Video:** Exponentially weighted averages
4. **Video:** Understanding exponentially weighted averages
5. **Video:** Bias correction in exponentially weighted averages
6. **Video:** Gradient descent with momentum
7. **Video:** RMSprop
8. **Video:** Adam optimization algorithm
9. **Video:** Learning rate decay
10. **Video:** The problem of local optima
11. **Notebook:** Optimization
12. **Video:** Yuanqing Lin interview

Show less

 **Graded:** Optimization algorithms

 **Graded:** Optimization

WEEK 3

Hyperparameter tuning, Batch Normalization and Programming Frameworks

 11 videos

1. **Video:** Tuning process
2. **Video:** Using an appropriate scale to pick hyperparameters
3. **Video:** Hyperparameters tuning in practice: Pandas vs. Caviar
4. **Video:** Normalizing activations in a network

5. **Video:** Fitting Batch Norm into a neural network
6. **Video:** Why does Batch Norm work?
7. **Video:** Batch Norm at test time
8. **Video:** Softmax Regression
9. **Video:** Training a softmax classifier
10. **Video:** Deep learning frameworks
11. **Video:** TensorFlow
12. **Notebook:** Tensorflow

Show less



Graded: Hyperparameter tuning, Batch Normalization, Programming Frameworks



Graded: Tensorflow

View Less

How It Works

GENERAL

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.

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, 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.

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.

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.

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.

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.

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.

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

Course 2 of Specialization

Deep Learning Specialization

Master Deep Learning, and Break into AI

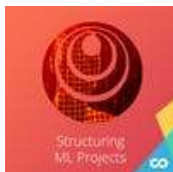


Deep Learning

deeplearning.ai

[View the course in catalog](#)

Related Courses



Structuring Machine Learning Projects

deeplearning.ai



Convolutional Neural Networks

deeplearning.ai

Sequence Models

deeplearning.ai



Neural Networks and Deep Learning

deeplearning.ai



Introduction to Deep Learning

National Research University Higher School of Economics

