

Deep Learning Adventures

Chapter 1 - Presentation 1

A quick overview of Coursera's Tensorflow in Practice specialization course

George Zoto

<https://www.meetup.com/Deep-Learning-Adventures>

In the beginning...

Meetup

Start a new group | Log in | Sign up

Deep Learning Adventures

Washington, DC · 160 members · Public group · Organized by George Z.

Share: [Facebook](#) [Twitter](#) [LinkedIn](#)

About Events Members Photos Discussions More

Join this group

Organizer

George Z. [Message](#)

Members (160) See all

FRI, APR 3, 7:30 PM

Deep Learning Adventures - Chapter 1

Online event

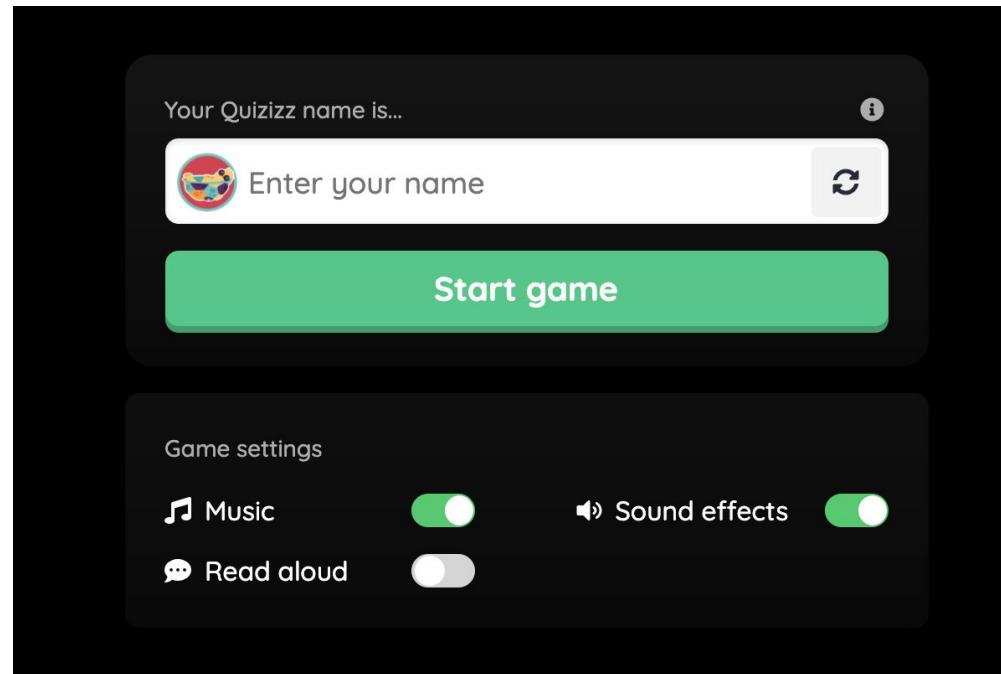
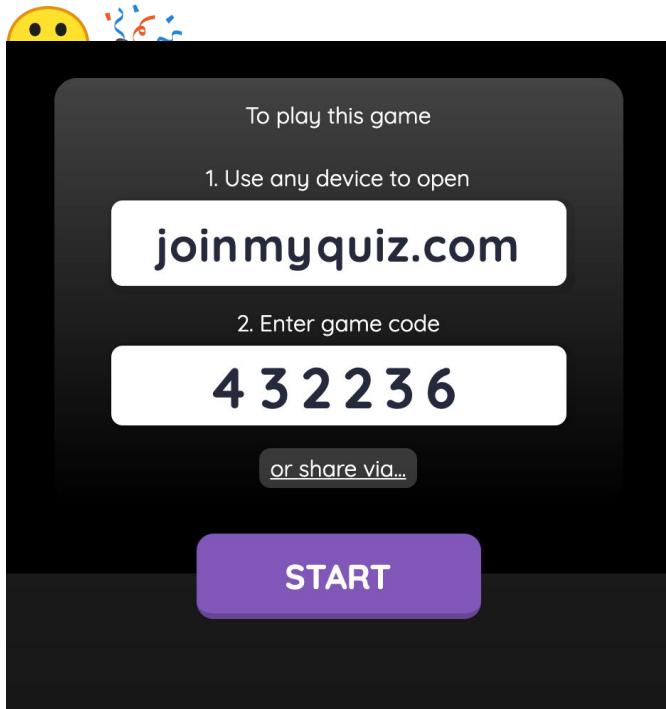
Join us for our very first adventure in Deep Learning! Just bring your curiosity with you and be ready to meet our growing community 😊 Agenda: - Introductions - What is this cool meetup about? - Industry updates and other interesting meetups...

- Meetup Link
<https://www.meetup.com/Deep-Learning-Adventures>
- GitHub repository
<https://github.com/georgezoto/Deep-Learning-Adventures>
- Join us on Slack
https://join.slack.com/t/deeplearninga-nmk8930/shared_invite/zt-d52h9mm9-h~Q0ZXw5PXsTDzPlINlvog
- YouTube recording of this cool event

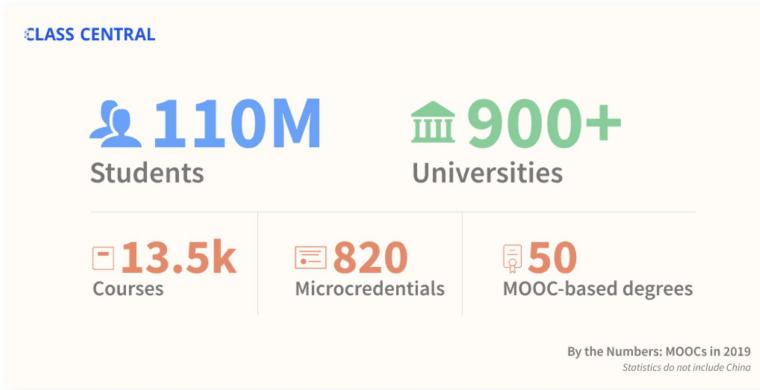


<https://youtu.be/wMujYXCz6pw>

Not a typical Meetup... Get ready for a fun game



MOOCs, Coursera, TensorFlow



	Learners	Courses	Microcredentials	Degrees
<u>Coursera</u>	45 million	3,800	420	16
<u>edX</u>	24 million	2,640	292	10
<u>Udacity</u>	11.5 million	200	40	1
<u>FutureLearn</u> ^{2,4}	10 million	880	49	23
<u>Swayam</u> ^{2,3}	10 million	1,000	0	0

Source: <https://www.classcentral.com/report/mooc-stats-2019/>

TensorFlow: An end-to-end open source machine learning platform

[TensorFlow](#) [Install](#) [Learn](#) [API](#) [Resources](#) [Community](#) [Why TensorFlow](#) [Search](#) [English](#) [GitHub](#) [Sign in](#)

Solutions to common ML problems

Simple step-by-step walkthroughs to solve common ML problems with TensorFlow.



For beginners

Your first neural network

Train a neural network to classify images of clothing, like sneakers and shirts, in this fast-paced overview of a complete TensorFlow program.



For experts

Generative adversarial networks

Train a generative adversarial network to generate images of handwritten digits, using the Keras Subclassing API.

Find | Compare

TensorFlow
Software

Keras
Search term

PyTorch
Computer application

+ Add comparison

United States

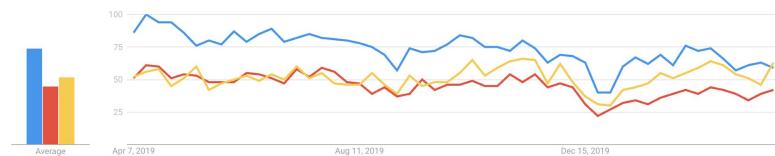
Past 12 months

All categories

Web Search

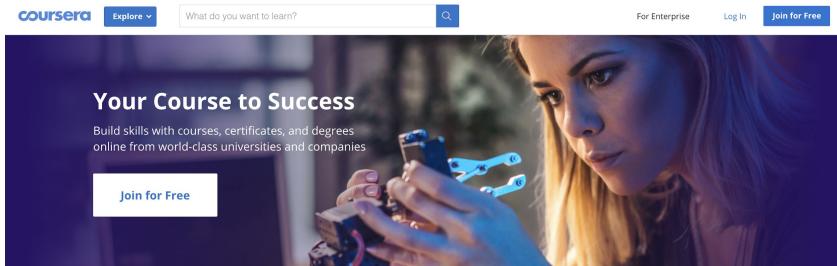
Note: This comparison contains both Search terms and Topics, which are measured differently. [LEARN MORE](#)

Interest over time



Source: <https://www.tensorflow.org/>

Attribution to Coursera and deeplearning.ai



A screenshot of the deeplearning.ai website homepage. The header features the deeplearning.ai logo and a navigation menu with links for "Courses", "Workers", "The Batch", "Events", "Forums", "Blog", and "Company". The main section has a dark background with the text "Break Into AI" in large white letters. Below it, a paragraph reads: "Whether you want to build algorithms or build a company, deeplearning.ai's courses will teach you key concepts and applications of AI." A red button at the bottom says "Take the Deep Learning Specialization". To the right, there's a graphic of a laptop screen showing two images: a "Content Image" of the Golden Gate Bridge and a "Generated Image" of the same bridge in a painterly style, labeled "Art Generation with Deep Learning".

Source:

<https://www.coursera.org/about/terms>
<https://www.coursera.org/>
<https://www.deeplearning.ai/>

Chapter 1 - TensorFlow in Practice Specialization

About this Specialization

199,621 recent views

Discover the tools software developers use to build scalable AI-powered algorithms in TensorFlow, a popular open-source machine learning framework.

In this four-course Specialization, you'll explore exciting opportunities for AI applications. Begin by developing an understanding of how to build and train neural networks. Improve a network's performance using convolutions as you train it to identify real-world images. You'll teach machines to understand, analyze, and respond to human speech with natural language processing systems. Learn to process text, represent sentences as vectors, and input data to a neural network. You'll even train an AI to create original poetry!

AI is already transforming industries across the world. After finishing this Specialization, you'll be able to apply your new TensorFlow skills to a wide range of problems and projects.

Looking for more advanced TensorFlow content? Check out the new [TensorFlow: Data and Deployment Specialization](#).

Chapter 1 - TensorFlow in Practice Specialization

There are 4 Courses in this Specialization

COURSE

1

Introduction to TensorFlow for Artificial Intelligence, Machine Learning, and Deep Learning

★★★★★ 4.7 6,196 ratings • 1,282 reviews

If you are a software developer who wants to build scalable AI-powered algorithms, you need to understand how to use the tools to build them. This course is part of the upcoming Machine Learning in Tensorflow Specialization and will teach you best practices for using TensorFlow, a popular open-source framework for machine learning.

[SHOW ALL](#)



6 hours to complete

A New Programming Paradigm

Welcome to this course on going from Basics to Mastery of TensorFlow. We're excited you're here! In week 1 you'll get a soft introduction to what Machine Learning and Deep Learning are, and how they offer you a new programming paradigm, giving you a new set of tools to open previously unexplored scenarios. All you need to know is some very basic SHOW ALL



4 videos (Total 16 min), 5 readings, 3 quizzes [SEE ALL](#)

COURSE

2

Convolutional Neural Networks in TensorFlow

★★★★★ 4.7 2,751 ratings • 410 reviews

If you are a software developer who wants to build scalable AI-powered algorithms, you need to understand how to use the tools to build them. This course is part of the upcoming Machine Learning in Tensorflow Specialization and will teach you best practices for using TensorFlow, a popular open-source framework for machine learning.

[SHOW ALL](#)



7 hours to complete

Introduction to Computer Vision

Welcome to week 2 of the course! In week 1 you learned all about how Machine Learning and Deep Learning is a new programming paradigm. This week you're going to take that to the next level by beginning to solve problems of computer vision with just a few lines of code! SHOW ALL



7 videos (Total 15 min), 6 readings, 3 quizzes [SEE ALL](#)

COURSE

3

Natural Language Processing in TensorFlow

★★★★★ 4.6 2,037 ratings • 277 reviews

If you are a software developer who wants to build scalable AI-powered algorithms, you need to understand how to use the tools to build them. This Specialization will teach you best practices for using TensorFlow, a popular open-source framework for machine learning.

[SHOW ALL](#)



8 hours to complete

Enhancing Vision with Convolutional Neural Networks

Welcome to week 3! In week 2 you saw a basic Neural Network for Computer Vision. It did the job nicely, but it was a little naive in its approach. This week we'll see how to make it better, as discussed by Laurence and Andrew here. SHOW ALL



6 videos (Total 19 min), 6 readings, 3 quizzes [SEE ALL](#)

COURSE

4

Sequences, Time Series and Prediction

★★★★★ 4.6 1,374 ratings • 223 reviews

If you are a software developer who wants to build scalable AI-powered algorithms, you need to understand how to use the tools to build them. This Specialization will teach you best practices for using TensorFlow, a popular open-source framework for machine learning.

[SHOW ALL](#)



9 hours to complete

Using Real-world Images

Last week you saw how to improve the results from your deep neural network using convolutions. It was a good start, but the data you used was very basic. What happens when your images are larger, or if the features aren't always in the same place? Andrew and Laurence discuss this to prepare you for what you'll learn this week: handling complex images! SHOW ALL

Source: <https://www.coursera.org/specializations/tensorflow-in-practice>

Setup



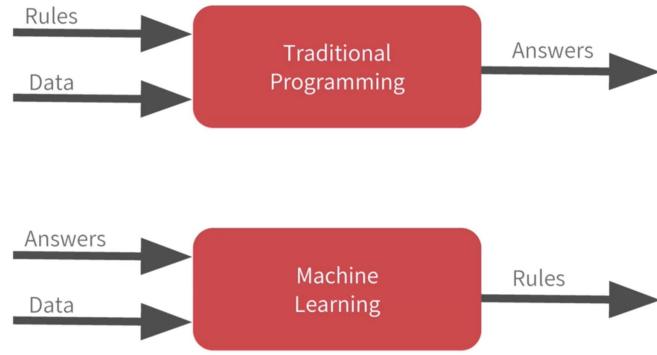
Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. You can write and execute code, save and share your analyses, and access powerful computing resources, all for free from your browser.

<https://colab.research.google.com>

Course 1: Introduction to TensorFlow for Artificial Intelligence, Machine Learning, and Deep Learning

Week 1: A New Programming Paradigm

A New Programming Paradigm



Activity Recognition



```
if(speed<4){  
    status=WALKING;  
}  
  
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}  
  
if(speed<4){  
    status=WALKING;  
} else if(speed<12){  
    status=RUNNING;  
} else {  
    status=BIKING;  
}  
  
// Oh crap
```

A New Programming Paradigm

Activity Recognition



0101001010100101010
1001010101001011101
0100101010010101001
0101001010100101010

Label = WALKING



1010100101001010101
01010100100100001
00100111101010111
1010100100111101011

Label = RUNNING



1001010011111010101
1101010110010101110
0010011110101011111
1010101111010101011

Label = BIKING



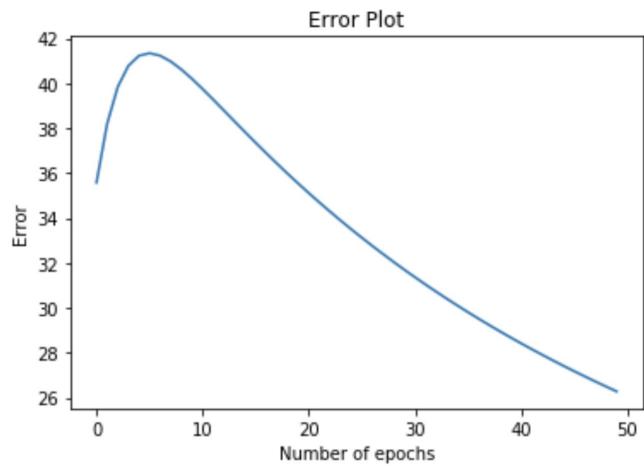
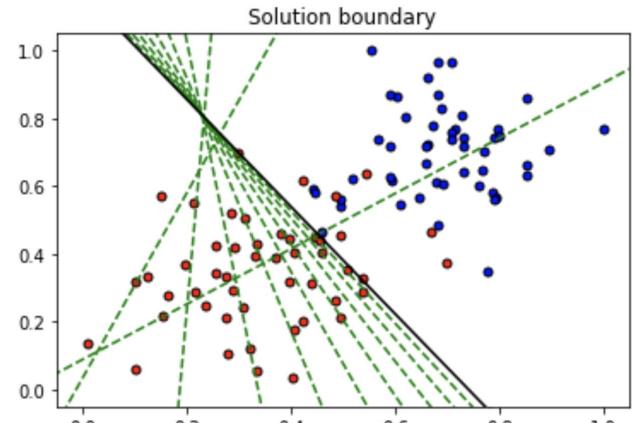
1111111111010011101
0011111010111110101
0101110101010101110
1010101010101011110

Label = GOLFING
(Sort of)



$$X = -1, 0, 1, 2, 3, 4$$

$$Y = -3, -1, 1, 3, 5, 7$$



A New Programming Paradigm

```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)

print(model.predict([10.0]))
```

A New Programming Paradigm

▼ The Hello World of Deep Learning with Neural Networks

Like every first app you should start with something super simple that shows the overall scaffolding for how your code works.

In the case of creating neural networks, the sample I like to use is one where it learns the relationship between two numbers. So, for example, if you were writing code for a function like this, you already know the 'rules' –

```
float hw_function(float x){  
    float y = (2 * x) - 1;  
    return y;  
}
```

So how would you train a neural network to do the equivalent task? Using data! By feeding it with a set of Xs, and a set of Ys, it should be able to figure out the relationship between them.

This is obviously a very different paradigm than what you might be used to, so let's step through it piece by piece.

▼ Imports

Let's start with our imports. Here we are importing TensorFlow and calling it tf for ease of use.

We then import a library called numpy, which helps us to represent our data as lists easily and quickly.

The framework for defining a neural network as a set of Sequential layers is called keras, so we import that too.

```
[2] import tensorflow as tf  
import numpy as np  
from tensorflow import keras
```

```
[3] print(tf.__version__)
```

```
↳ 2.2.0-rc2
```

```
[4] print(keras.__version__)
```

```
↳ 2.3.0-tf
```

▼ Define and Compile the Neural Network

Next we will create the simplest possible neural network. It has 1 layer, and that layer has 1 neuron, and the input shape to it is just 1 value.

```
[5] model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
```

```
▶ model.summary()
```

```
↳ Model: "sequential"
```

Layer (type)	Output Shape	Param #
<hr/>		
dense (Dense)	(None, 1)	2
<hr/>		
Total params:	2	
Trainable params:	2	
Non-trainable params:	0	

A New Programming Paradigm

Over time you will learn the different and appropriate loss and optimizer functions for different scenarios.

```
[7] model.compile(optimizer='sgd', loss='mean_squared_error')
```

▼ Providing the Data

Next up we'll feed in some data. In this case we are taking 6 xs and 6ys. You can see that the relationship between these is that $y=2x-1$, so where $x = -1, y=-3$ etc. etc.

A python library called 'Numpy' provides lots of array type data structures that are a defacto standard way of doing it. We declare that we want to use these by specifying the values as an np.array[]

```
[10] xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
      ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)
```

A New Programming Paradigm

• Training the Neural Network

The process of training the neural network, where it 'learns' the relationship between the Xs and Ys is in the `model.fit` call. This is where it will go through the loop we spoke about above, making a guess, measuring how good or bad it is (aka the loss), using the optimizer to make another guess etc. It will do it for the number of epochs you specify. When you run this code, you'll see the loss on the right hand side.

```
[13] model.fit(xs, ys, epochs=300)

→ Epoch 1/300
1/1 [=====] - 0s 2ms/step - loss: 0.2848
Epoch 2/300
1/1 [=====] - 0s 1ms/step - loss: 0.2789
Epoch 3/300
1/1 [=====] - 0s 3ms/step - loss: 0.2732
Epoch 4/300
1/1 [=====] - 0s 3ms/step - loss: 0.2676
Epoch 5/300
1/1 [=====] - 0s 1ms/step - loss: 0.2621
Epoch 6/300
1/1 [=====] - 0s 1ms/step - loss: 0.2567
Epoch 7/300
1/1 [=====] - 0s 1ms/step - loss: 0.2514
```

A New Programming Paradigm

```
1/1 [=====] - 0s 2ms/step - loss: 5.8673e-04
Epoch 300/300
1/1 [=====] - 0s 2ms/step - loss: 5.7468e-04
<tensorflow.python.keras.callbacks.History at 0x7fe71cb594e0>
```

Ok, now you have a model that has been trained to learn the relationship between X and Y. You can use the **model.predict** method to have it figure out the Y for a previously unknown X. So, for example, if X = 10, what do you think Y will be? Take a guess before you run this code:

```
[14] print(model.predict([10.0]))
```

```
⇒ [[18.93006]]
```

You might have thought 19, right? But it ended up being a little under. Why do you think that is?

A New Programming Paradigm

That brings you to the end of what you need to look at for Week 1. If you're eager to learn more, before we go to Week 2, there are some great resources you can check out:

- AI For Everyone is a non-technical course that will help you understand many of the AI technologies we will discuss later in this course, and help you spot opportunities in applying this technology to solve your problems.
<https://www.deeplearning.ai/ai-for-everyone/>
- TensorFlow is available at [TensorFlow.org](https://www.tensorflow.org), and video updates from the TensorFlow team are at youtube.com/tensorflow

Play with a neural network right in the browser at <http://playground.tensorflow.org>. See if you can figure out the parameters to get the neural network to pattern match to the desired groups. The spiral is particularly challenging!

The 'Hello World' notebook that we used in this course is available on GitHub [here](#).

A New Programming Paradigm

<https://github.com/lmoroney/dlaicourse>

Watch 181 ⚡ Star 2,580 Fork 2.4k

Code Issues 33 Pull requests 24 Actions Projects 0 Wiki Security Insights

Notebooks for learning deep learning

- 240 commits 2 branches 0 packages 0 releases 9 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

lmoroney Added Colab Link

Latest commit 8b6dd9a yesterday

File	Commit Message	Time Ago
Exercises	more =>	6 days ago
TensorFlow Deployment	more =>	6 days ago
TensorFlow In Practice	Added Colab Link	yesterday
Course 1 - Part 2 - Lesson 2 - Notebook.ipynb	nbfmft	6 days ago
Course 1 - Part 4 - Lesson 2 - Notebook.ipynb	tf.optimizers.Adam	6 days ago
Course 1 - Part 4 - Lesson 4 - Notebook.ipynb	accuracy, adam	6 days ago
Course 1 - Part 6 - Lesson 2 - Notebook.ipynb	nbfmft	6 days ago
Course 1 - Part 6 - Lesson 3 - Notebook.ipynb	nbfmft	6 days ago
Course 1 - Part 8 - Lesson 2 - Notebook.ipynb	acc => accuracy	6 days ago
Course 1 - Part 8 - Lesson 3 - Notebook.ipynb	acc => accuracy	6 days ago
Course 1 - Part 8 - Lesson 4 - Notebook.ipynb	acc => accuracy	6 days ago
Course 2 - Part 2 - Lesson 2 - Notebook.ipynb	acc => accuracy	6 days ago
Course 2 - Part 4 - Lesson 2 - Notebook.ipynb	acc => accuracy	6 days ago
Course 2 - Part 6 - Lesson 3 - Notebook.ipynb	acc => accuracy	6 days ago
Course 2 - Part 8 - Lesson 2 - Notebook (Cats v Dogs Augmentation).ipynb	acc => accuracy	6 days ago
Course 2 - Part 4 - Lesson 4 - Notebook.ipynb	acc => accuracy	6 days ago
Hello_World_Layers.ipynb	nbfmft	6 days ago
Horse_or_Human_WithDropouts.ipynb	acc => accuracy	6 days ago
Horse_or_Human_NoValidation.ipynb	acc => accuracy	6 days ago

<https://github.com/lmoroney/dlaicourse>

Watch 183 ⚡ Star 2.6k Fork 2.4k

Code Issues 33 Pull requests 26 Actions Projects 0 Wiki Security Insights

Branch: master dlaicourse / Course 1 - Part 2 - Lesson 2 - Notebook.ipynb Find file Copy path

MarkDaoust nbfmft b8d7f9e 8 days ago

4 contributors

285 lines (285 sloc) | 10.2 KB

Open in Colab

Copyright 2019 The TensorFlow Authors.

```
In [0]: #title Licensed under the Apache License, Version 2.0 (the "License");
# You may not use this file except in compliance with the License.
#
# You may obtain a copy of the License at
#
# https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

The Hello World of Deep Learning with Neural Networks

Like every first app you should start with something super simple that shows the overall scaffolding for how your code works.

In the case of creating neural networks, the sample I like to use is one where it learns the relationship between two numbers. So, for example, if you were writing code for a function like this, you already know the 'rules' –

```
float hw_function(float x){
    float y = (2 * x) - 1;
    return y;
}
```

So how would you train a neural network to do the equivalent task? Using data! By feeding it with a set of Xs, and a set of Ys, it should be able to figure out the relationship between them.

This is obviously a very different paradigm than what you might be used to, so let's step through it piece by piece.

Source: <https://github.com/lmoroney/dlaicourse>

Check out these resources

AI For Everyone

AI is not only for engineers. “AI for Everyone”, a non-technical course, will help you understand AI technologies and spot opportunities to apply AI to problems in your own organization. You will see examples of what today’s AI can – and cannot – do. Finally, you will understand how AI is impacting society and how to navigate through this technological change.

If you are a non-technical business professional, “AI for Everyone” will help you understand how to build a sustainable AI strategy. If you are a machine learning engineer or data scientist, this is the course to ask your manager, VP or CEO to take if you want them to understand what you can (and cannot!) do.



Week 1
What is AI



Week 2
Building AI Projects



Week 3
AI in Your Company



Week 4
AI and Society

Source: <https://www.deeplearning.ai/ai-for-everyone/>

Check out these resources

 **TensorFlow**
221K subscribers

[HOME](#) [VIDEOS](#) [PLAYLISTS](#) [COMMUNITY](#) [CHANNELS](#) [ABOUT](#) 

TensorFlow Developer Summit 2020 ► [PLAY ALL](#)

Thanks for joining us in our first-ever virtual-only #TFDevSummit. If you missed out on any of the content, check out all the sessions in this playlist! The content focuses on TensorFlow updates for

TensorFlow Dev Summit 2020 Livestream
TensorFlow • 34K views • Streamed 3 weeks ago
Mark your calendar for #TFDevSummit happening on March 11 and join TensorFlow users from all over the world. The content focuses on TensorFlow updates for researchers, production scaling,
[CC](#)

TensorFlow Dev Summit 2020 Keynote
TensorFlow • 18K views • 3 weeks ago
Join the TensorFlow team as they kick-off the 2020 TensorFlow Dev Summit. The keynote will feature new product updates for the TensorFlow ecosystem. Speakers: Megan Kacholia - VP, Engineering
[CC](#)

Learning to read with TensorFlow and Keras (TF Dev Summit '20)
TensorFlow • 12K views • 2 weeks ago
Natural Language Processing (NLP) has hit an inflection point, and this talk shows you how TensorFlow and Keras make it easy to preprocess, train, and hypertune text models. Speaker: Paige Bailey ...
[CC](#)

TensorFlow Lite: ML for mobile and IoT devices (TF Dev Summit '20)
TensorFlow • 6.6K views • 2 weeks ago
Learn about how to deploy ML to mobile phones and embedded devices. Now deployed on billions of devices in production - it's the world's best cross-platform ML framework for mobile and microcontrol...
[CC](#)

[SHOW MORE](#)

Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.

Epoch 000,000 Learning rate 0.03 Activation Tanh Regularization None Regularization rate 0 Problem type Classification

DATA
Which dataset do you want to do?

Ratio of training to test data: 50%
Noise: 0
Batch size: 10
[REGENERATE](#)

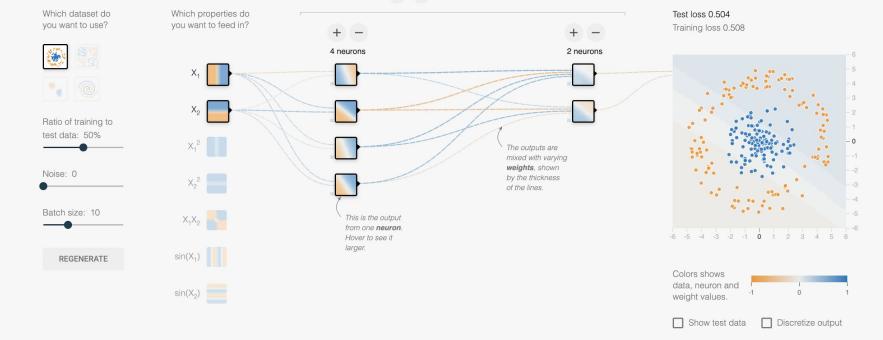
FEATURES
Which properties do you want to feed in?
 x_1 x_2 x_1^2 x_2^2 x_1x_2 $\sin(x_1)$ $\sin(x_2)$

HIDDEN LAYERS
+ - 2 HIDDEN LAYERS
+ - 4 neurons
+ - 2 neurons

OUTPUT
Test loss 0.504 Training loss 0.508

The outputs are mixed with varying weights, shown by the thickness of the lines.
This is the output from one neuron. Hover to see it larger.

Colors show data, neuron and weight values.
 Show test data Discretize output



Source:

<https://www.youtube.com/tensorflow>
<http://playground.tensorflow.org/>

Check out these events and Meetups

you learn about building machine learning models for diagnosis.

0:11 / 1:28

New AI For Medicine Specialization coming soon!

Thursday, March 26, 2020

Online: Using Wikipedia and Wikidata for Natural Language Processing

Hosted by
Seth Grimes

Saturday, March 28, 2020

AI Meet & Greet



Hosted by
viraf and 2 others

Thursday, April 2, 2020

Virtual Event : Session 3 - [Udacity] Intro to Deep Learning using PyTorch



Hosted by
Pragyansmita Nayak, Ph.D and Data Community DC (DC2)

Thursday, April 2, 2020

Remote Session: Attention Is All You Need



Hosted by
Julius

Source:

https://www.youtube.com/watch?v=zGFKSQLef_0

<https://www.meetup.com/DC-NLP/events/269502685>

<https://www.meetup.com/Bethesda-Artificial-Intelligence-Meetup>

<https://www.meetup.com/novadeeplearning>

<https://www.meetup.com/Machine-Learning-Paper-Club>

Check out this new certification and online conference

TensorFlow Core

Introducing the TensorFlow Developer Certificate!

March 12, 2020



Posted by Alina Shinkarsky, on behalf of the TensorFlow Team

In the AI world today, more and more companies are looking to hire machine learning talent, and simultaneously, an increasing number of students and developers are looking for ways to gain and showcase their ML knowledge with formal recognition. In addition to the courses and learning resources available online, we want to help developers showcase their ML proficiency and help companies hire ML developers to solve challenging problems.



Workshop Outline

TOPIC	DESCRIPTION
Introduction (15 mins)	<ul style="list-style-type: none">> Meet the instructor.> Create an account at courses.nvidia.com/join
Image Segmentation with TensorFlow (120 mins)	<ul style="list-style-type: none">> Compare image segmentation to other computer vision problems.> Experiment with TensorFlow tools.> Implement effective metrics for assessing model performance.
Break (60 mins)	
Word Generation with TensorFlow (120 mins)	<ul style="list-style-type: none">> Learn about natural language processing (NLP) and recurrent neural networks (RNNs).> Create network inputs from text data.> Test with new data and iterate to improve performance.
Break (15 mins)	
Image and Video Captioning (120 mins)	<ul style="list-style-type: none">> Combine computer vision and natural language processing to describe scenes.> Learn to harness the functionality of convolutional neural networks (CNNs) and RNNs.
Final Review (15 mins)	<ul style="list-style-type: none">> Review key learnings and wrap up questions.> Complete the assessment to earn a certificate.> Take the workshop survey.

Source:

<https://blog.tensorflow.org/2020/03/introducing-tensorflow-developer-certificate.html?m=1>
<https://www.nvidia.com/en-us/gtc/>

Time for a fun game



To play this game

1. Use any device to open
joinmyquiz.com
2. Enter game code
432236

or share via...

START

Your Quizizz name is... i ↻

Start game

Game settings

Music Sound effects
Read aloud

Practice here or use Flashcards:

quizizz.com/join/quiz/5e87bdbc07fa7f001b120404/start?from=sol_oLinkShare&referrer=5d921444d0fa99001a135336

Time for a fun game



Practice here or use Flashcards:

quizizz.com/join/quiz/5e87bdb07fa7f001b120404/start?from=soloLinkShare&referrer=5d921444d0fa99001a135336

The image shows two side-by-side screenshots. On the left is the Quizizz Live Dashboard for a game with the code 432 236. It displays a list of 22 players with their names, profile icons, scores, and a 'Leave' button. A video feed of a person is visible in the top right corner. On the right is a slide from a Google Slides presentation titled 'Deep-Learning-Adventures-Chapter-1-Presentation-1'. The slide has a purple header with the text 'How do you import Tensorflow in your Python code'. Below the header are four numbered options: 1. 'import tf from ai', 2. 'import tensorflow from google', 3. 'expot tensorflow as tf', and 4. 'import tensorflow as tf'. The slide also includes navigation controls like a play button, a progress bar showing 4/12, and a timer at 100m.

Player	Name	Score
6	Anil	8630
6	rek	8630
7	Charles Stockman	7450
8	Martin	7290
9	Angel	7230
10	Chuba	7050
11	Peter	6880
12	Dean	6840
13	Melissa	6505
14	Sanjay	5710
15	Hasson	4730
16	AH	4470
16	Kottie	4470
17	v	4010
18	KL	3910
19	Tony	3240
20	SS	2130
21	George	2100
22	Thomas	0

Questions

Discussion

2 Deep Learning representations

For representations:

- nodes represent inputs, activations or outputs
- edges represent weights or biases

Here are several examples of Standard deep learning representations

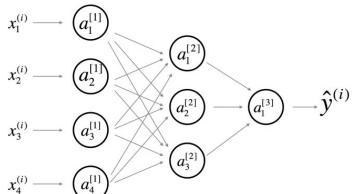


Figure 1: Comprehensive Network: representation commonly used for Neural Networks. For better aesthetic, we omitted the details on the parameters ($w_{ij}^{[l]}$ and $b_i^{[l]}$ etc...) that should appear on the edges

Standard notations for Deep Learning

This document has the purpose of discussing a new standard for deep learning mathematical notations.

1 Neural Networks Notations.

General comments:

- superscript (i) will denote the i^{th} training example while superscript [l] will denote the l^{th} layer

Sizes:

$\cdot m$: number of examples in the dataset

$\cdot n_x$: input size

$\cdot n_y$: output size (or number of classes)

$\cdot n_h^{[l]}$: number of hidden units of the l^{th} layer

In a for loop, it is possible to denote $n_x = n_h^{[0]}$ and $n_y = n_h^{[\text{number of layers} + 1]}$.

$\cdot L$: number of layers in the network.

Objects:

$\cdot X \in \mathbb{R}^{n_x \times m}$ is the input matrix

$\cdot x^{(i)} \in \mathbb{R}^{n_x}$ is the i^{th} example represented as a column vector

$\cdot Y \in \mathbb{R}^{n_y \times m}$ is the label matrix

$\cdot y^{(i)} \in \mathbb{R}^{n_y}$ is the output label for the i^{th} example

$\cdot W^{[l]} \in \mathbb{R}^{\text{number of units in next layer} \times \text{number of units in the previous layer}}$ is the weight matrix, superscript [l] indicates the layer

$\cdot b^{[l]} \in \mathbb{R}^{\text{number of units in next layer}}$ is the bias vector in the l^{th} layer

$\cdot \hat{y} \in \mathbb{R}^{n_y}$ is the predicted output vector. It can also be denoted $a^{[L]}$ where L is the number of layers in the network.

Common forward propagation equation examples:

$a = g^{[l]}(W_x x^{(i)} + b_1) = g^{[l]}(z_1)$ where $g^{[l]}$ denotes the l^{th} layer activation function

$\hat{y}^{(i)} = \text{softmax}(W_h h + b_2)$

· General Activation Formula: $a_j^{[l]} = g^{[l]}(\sum_k w_{jk}^{[l]} a_k^{[l-1]} + b_j^{[l]}) = g^{[l]}(z_j^{[l]})$

· $J(x, W, b, y)$ or $J(\hat{y}, y)$ denote the cost function.

Examples of cost function:

$\cdot J_{CE}(\hat{y}, y) = -\sum_{i=0}^m y^{(i)} \log \hat{y}^{(i)}$

$\cdot J_1(\hat{y}, y) = \sum_{i=0}^m |y^{(i)} - \hat{y}^{(i)}|$

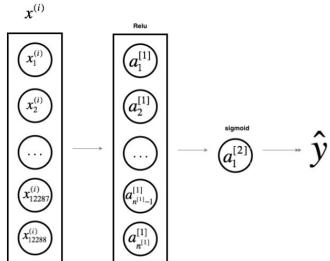


Figure 2: Simplified Network: a simpler representation of a two layer neural network, both are equivalent.

Content added after our 1st session, based on group feedback



George Zoto 1:53 PM

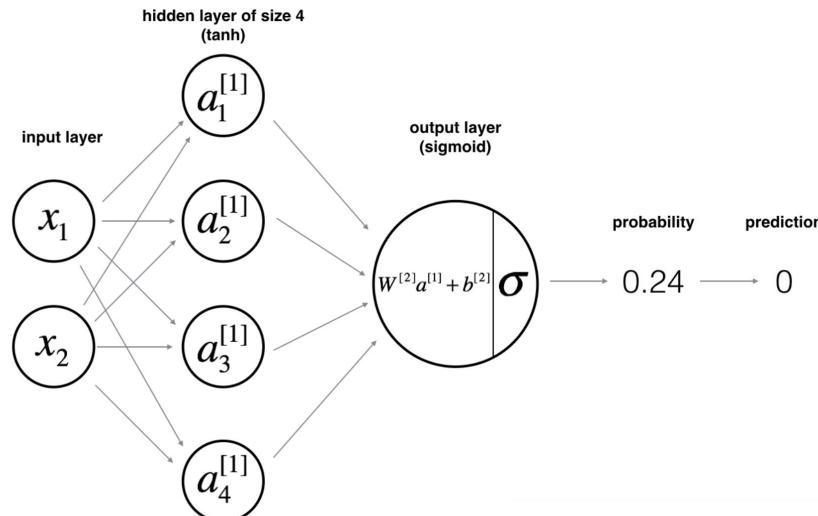
Hi friends! Looking at our comments last night about our first model and what is it really doing under the hood, I updated our code repository to include:

- A sample architecture of a 2 layer network (1 hidden + 1 output layer). Our model is even simpler than this, with just 1 neuron in the hidden layer.
- A custom callback function that is called during training
- A detailed chart plotted after each epoch showing how our model's weight (w) and bias (b) are adjusted by the optimizer (sgd) to minimize our loss function (mean_squared_error)

Sample architecture of a 2 layer network (1 hidden + 1 output layer)

It has 2 inputs/features (x_1 and x_2), 1 hidden layer of 4 nodes/neurons (a_1 through a_4) and final output layer of 1 node/neuron

Our model was even simpler than this, with just 1 neuron in the hidden layer as its only output layer.



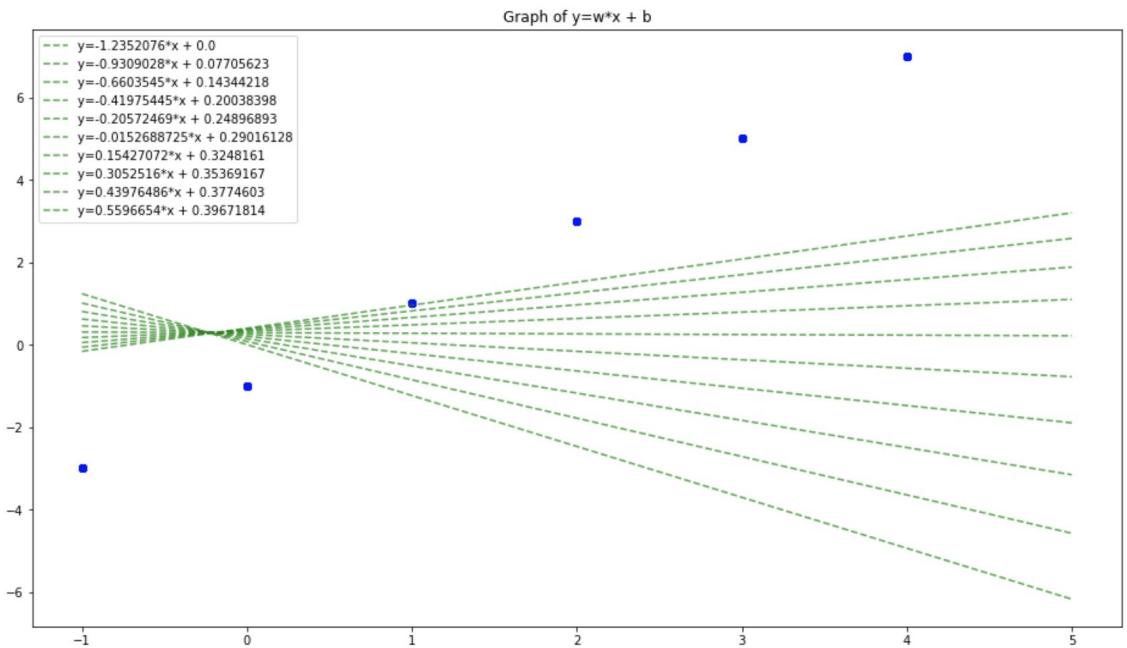
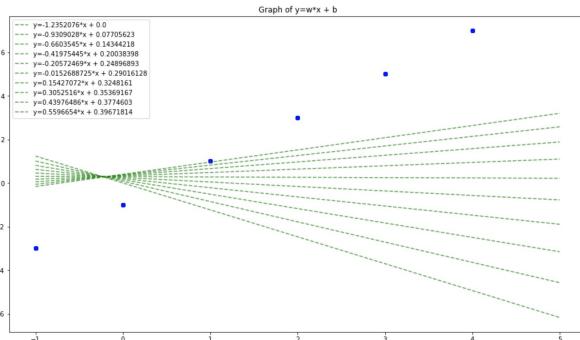
Our code is available on GitHub and on our Slack channel:
https://github.com/georgezoto/TensorFlow-in-Practice/blob/master/Course-1-Introduction-to-TensorFlow/C1W1_A_new_programming_paradigm_1.ipynb

#tensorflow-in-practice-specialization

You created this channel on March 31st. This is the very beginning of the #tensorflow-in-practice-specialization channel.

Content added after our 1st session, based on group feedback

```
: model.fit(xs, ys, epochs=10, callbacks=[get_weight_and_bias()])  
  
Epoch 1/10  
-1.2352076 0.0  
1/1 [=====] - 0s 2ms/step - loss: 45.3716  
Epoch 2/10  
-0.9309028 0.07705623  
1/1 [=====] - 0s 2ms/step - loss: 36.0725  
Epoch 3/10  
-0.6603545 0.14344218  
1/1 [=====] - 0s 2ms/step - loss: 28.7486  
Epoch 4/10  
-0.41975445 0.20038398  
1/1 [=====] - 0s 1ms/step - loss: 22.9789  
Epoch 5/10  
-0.20572469 0.24896893  
1/1 [=====] - 0s 1ms/step - loss: 18.4323  
Epoch 6/10  
-0.0152688725 0.29016128  
1/1 [=====] - 0s 1ms/step - loss: 14.8479  
Epoch 7/10  
0.15427072 0.3248161  
1/1 [=====] - 0s 1ms/step - loss: 12.0208  
Epoch 8/10  
0.3952516 0.35369167  
1/1 [=====] - 0s 1ms/step - loss: 9.7895  
Epoch 9/10  
0.43976486 0.3774603  
1/1 [=====] - 0s 1ms/step - loss: 8.0273  
Epoch 10/10  
0.5596654 0.39671814  
1/1 [=====] - 0s 1ms/step - loss: 6.6342  
: <tensorflow.python.keras.callbacks.History at 0x7f77aae00d0>
```



Our code is available on GitHub and on our Slack channel:
https://github.com/georgezoto/TensorFlow-in-Practice/blob/master/Course-1-Introduction-to-TensorFlow/C1W1_A_new_programming_paradigm_1.ipynb

#tensorflow-in-practice-specialization

You created this channel on March 31st. This is the very beginning of the #tensorflow-in-practice-specialization channel.