



Deep Learning

for Image Classification and Segmentation

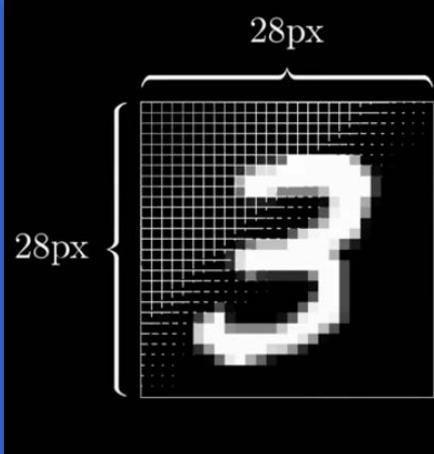
The study plan

- Unit 2: What is image classification and how to implement a solution?





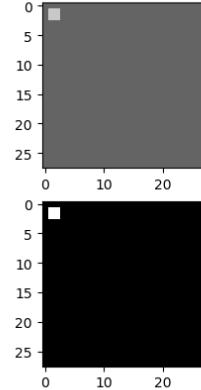
What is image classification?



- A gray image whose size is $n \times m$ has $n \times m$ pixels
- The gray level of each pixel is between 0 to 255 or 0 to 1 (normalized, double format)
- [How to convert a double matrix image to a unit8 matrix?](#)
- [How to convert a matrix to a Gray scale image?](#)
- Input an image and output a class label

What is image?

- Learning Python from comparing Matlab with Python



How to implement a solution?

- Using training images to train a Classifier
- Using testing images to evaluation the a Classifier
- Input an image and output a class label

Major Course Assignments

Individual Assignment Activity of Unit 3:

Run the Lab: Image Classification with DIGITS

Individual Assignment Activity of Unit 5: (option)

Run the Lab: Medical Image Segmentation with DIGITS

https://nvidia.qwiklab.com/users/sign_in?locale=en

<https://www.nvidia.com/en-us/deep-learning-ai/education/>





How to evaluate a classifier?

- Machines make mistake

Positive Negative

True TP NT

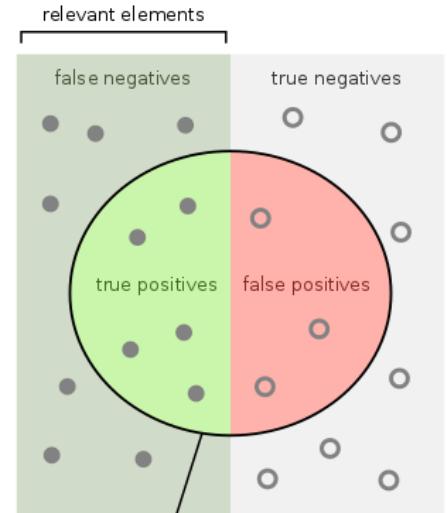
False FP FN

Concepts of Machine Learning

- Data acquisition and preprocessing
- Design Model
- Selecting Suitable Algorithm
- Parameters adjustment(training)
- Prediction
- Feedback results to modify parameters

Evaluation of Classifier

- True Positive (TP)
- True Negative (TN)
- False Positive (假正, FP)
- False Negative (假負 , FN)
- True Positive Rate (TPR) or (sensitivity)
$$TPR = TP / (TP + FN)$$
- True Negative Rate (TNR) or (specificity)
$$TNR = TN / (TN + FP)$$
- False Positive Rate (FPR)
$$FPR = FP / (FP + TN)$$
- False Negative Rate (FNR)
$$FNR = FN / (TP + FN)$$
- Source:
 - https://en.wikipedia.org/wiki/Sensitivity_and_specificity
 - https://en.wikipedia.org/wiki/False_positives_and_false_negatives



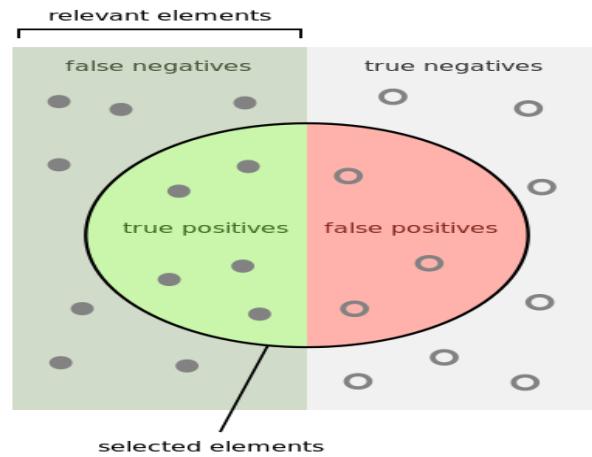
How many negative selected elements are truly negative?
e.g. How many healthy people are identified as not having the condition.

$$\text{Sensitivity} = \frac{\text{true positives}}{\text{relevant elements}}$$

$$\text{Specificity} = \frac{\text{true negatives}}{\text{relevant elements}}$$

Evaluation of Classifier

- True Positive (檢查有病也有病)



How many selected items are relevant?

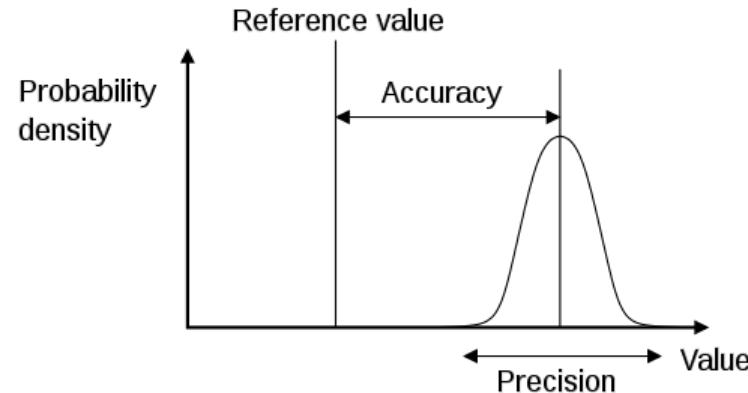
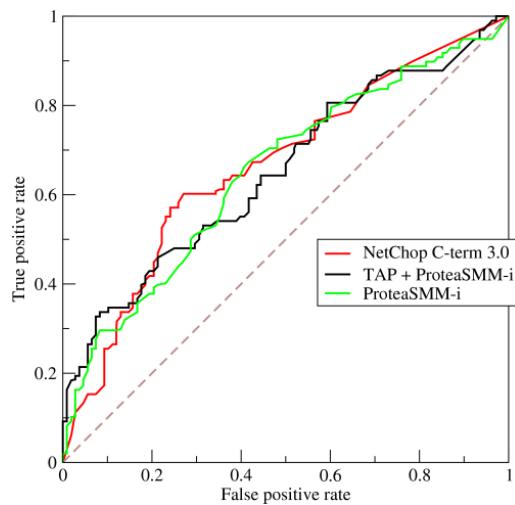
$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Evaluation of Classifier

- Receiver Operating Characteristic (ROC)
- Area under the Curve of ROC (AUC ROC))
- This curve plays a central role in evaluating diagnostic ability of tests to discriminate the true state of subjects, finding the optimal cut off values, and comparing two alternative diagnostic tasks when each task is performed on the same subject



Evaluation of Classifier

- Statistics
- Null and Alternative Hypothesis
- There are two types of errors:
 - Type I – H_0 is rejected even though it is true (**false positive**)
 - Type II – H_0 is not rejected even though it is false (**false negative**)

Evaluation of Classifier

- Significance level is the acceptable level of type I error, denoted α . Typically, a significance level of $\alpha = .05$ is used (although sometimes other levels such as $\alpha = .01$ may be employed).
- This means that we are willing to tolerate up to 5% of type I errors,
- i.e. we are willing to accept the fact that in 1 out of every 20 samples we reject the null hypothesis even though it is true.

Evaluation of Classifier

- P-value (the probability value) is the value p of the statistic used to test the null hypothesis. If $p < \alpha$ then we reject the null hypothesis.

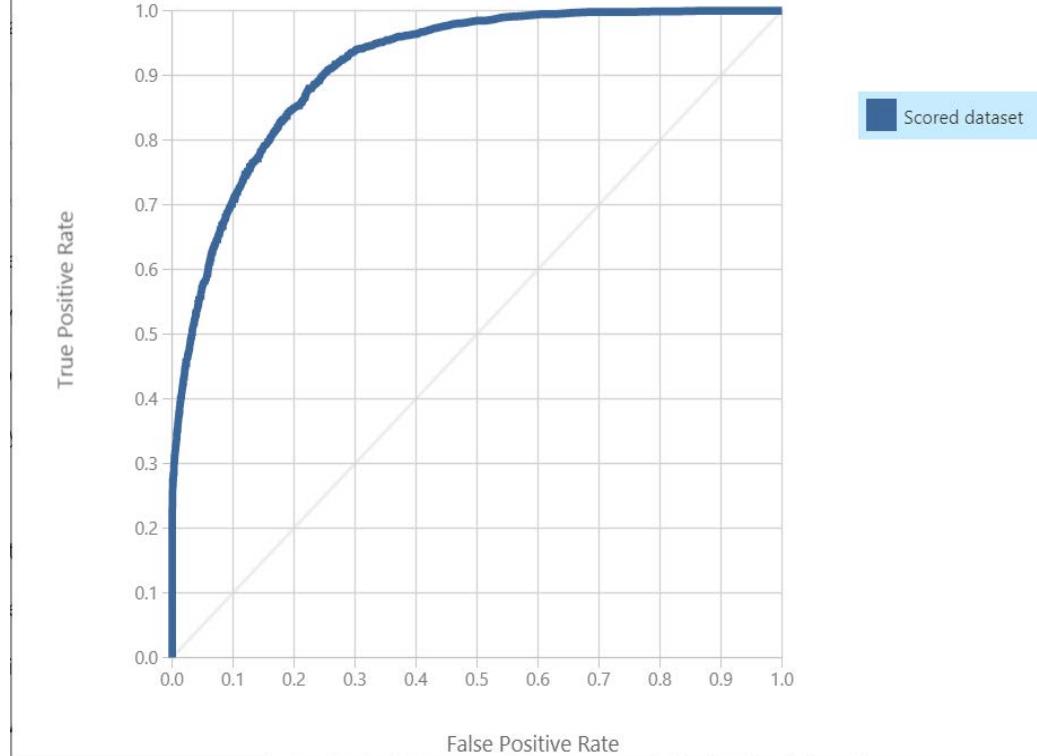
Azure Machine Learning

- Predict Annual Income
- Will somebody earn over 50k a year?
- Step 1: Getting the data
- Step 2: Select required columns
- Step 3: Clean missing data
- Step 4: Inspect the data
- Step 5: Take care of the variable types
- Step 6: Split the dataset into a training and a test set

Azure Machine Learning

- Step 7: Train the basic model
- Step 8: Evaluate the Feature Importance
- Step 9: Score the test data
- Step 10: Evaluate the model

ref.



Microsoft Power BI

- Machine learning tutorial: Create your first data science experiment in Azure Machine Learning Studio
- <https://docs.microsoft.com/en-us/azure/machine-learning/studio/create-experiment>

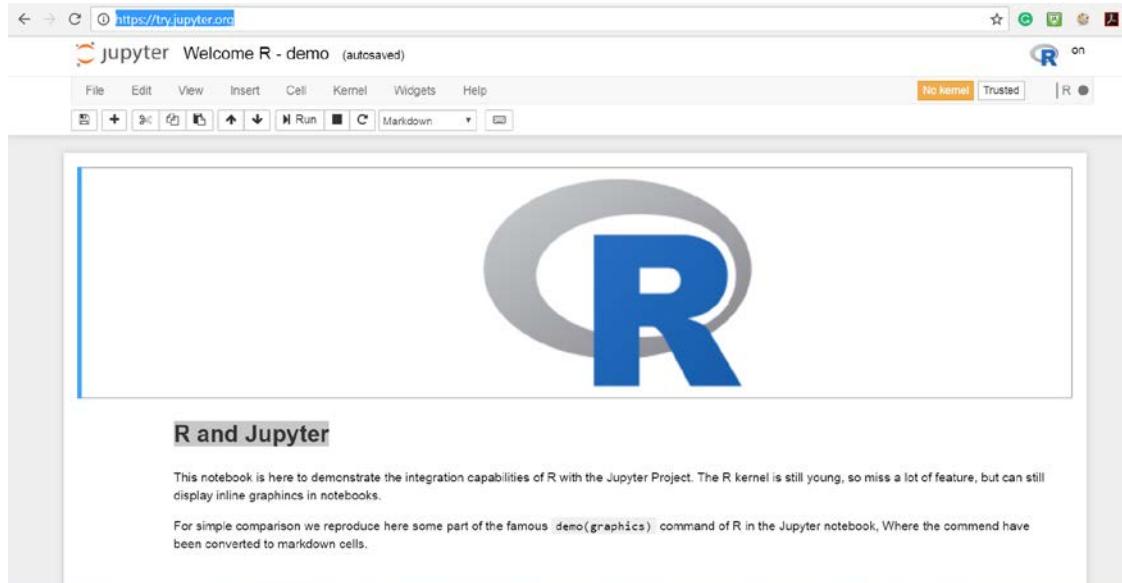
Five steps to create an experiment

In this machine learning tutorial, you'll follow five basic steps to build an experiment in Machine Learning Studio to create, train, and score your model:

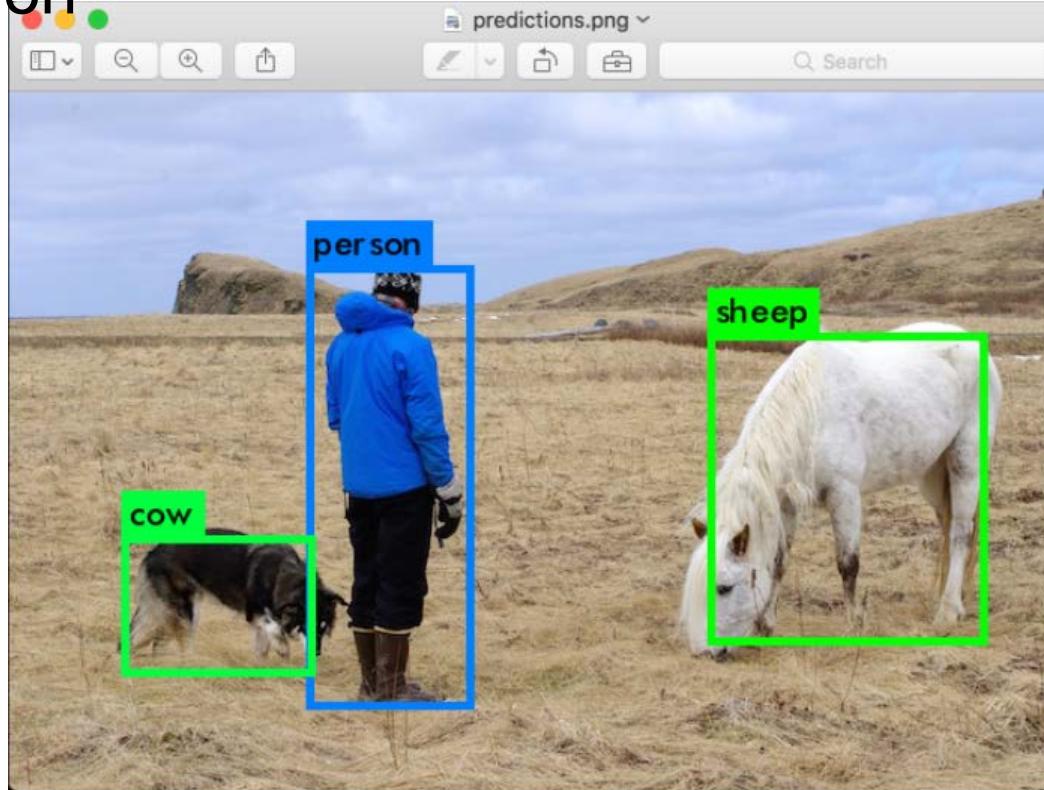
- **Create a model**
 - Step 1: Get data
 - Step 2: Prepare the data
 - Step 3: Define features
- **Train the model**
 - Step 4: Choose and apply a learning algorithm
- **Score and test the model**
 - Step 5: Predict new automobile prices

R and Jupyter

- <https://try.jupyter.org/>



Real-Time Object Detection & Recognition



Real-Time Detection

- Real-Time Detection on a Webcam
- Running YOLO on test data isn't very interesting if you can't see the result. Instead of running it on a bunch of images let's run it on the input from a webcam!
- To run this demo you will need to compile Darknet with CUDA and OpenCV. Then run the command:`./darknet detector demo cfg/coco.data cfg/yolo.cfg yolo.weights`

Image Classification

1. Nvidia lab
2. Training Experiment
3. Prediction Experiment

Image Classification with DIGITS

- Nvidia 雲端網站

https://nvidia.qwiklab.com/users/sign_in?locale=en

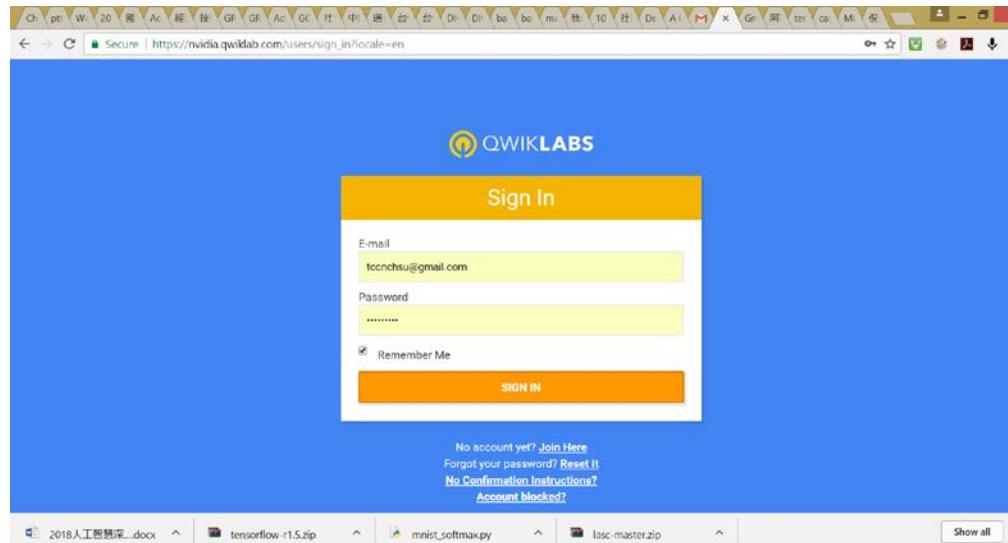


Image Classification with DIGITS

- Nvidia 雲端網站

https://nvidia.qwiklab.com/users/sign_in?locale=en

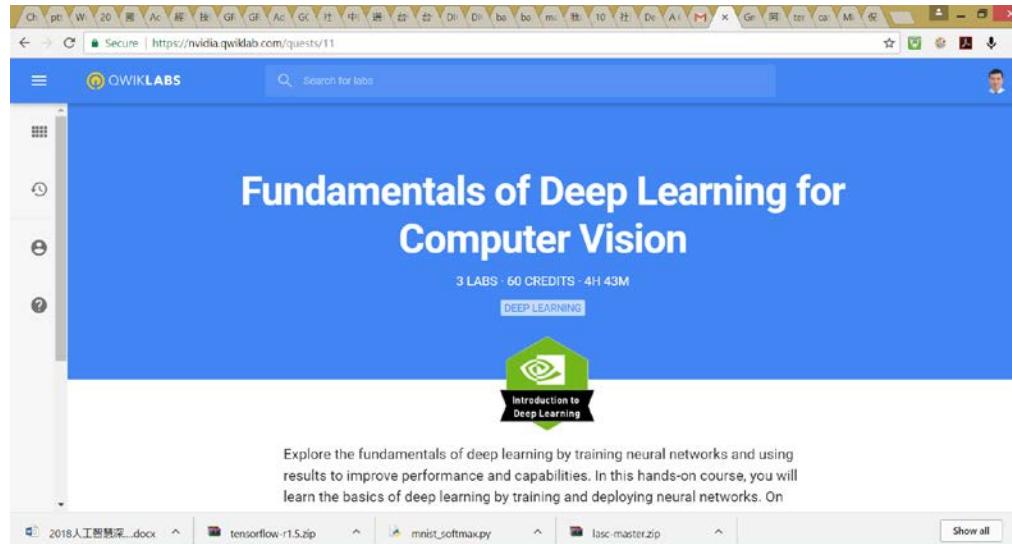
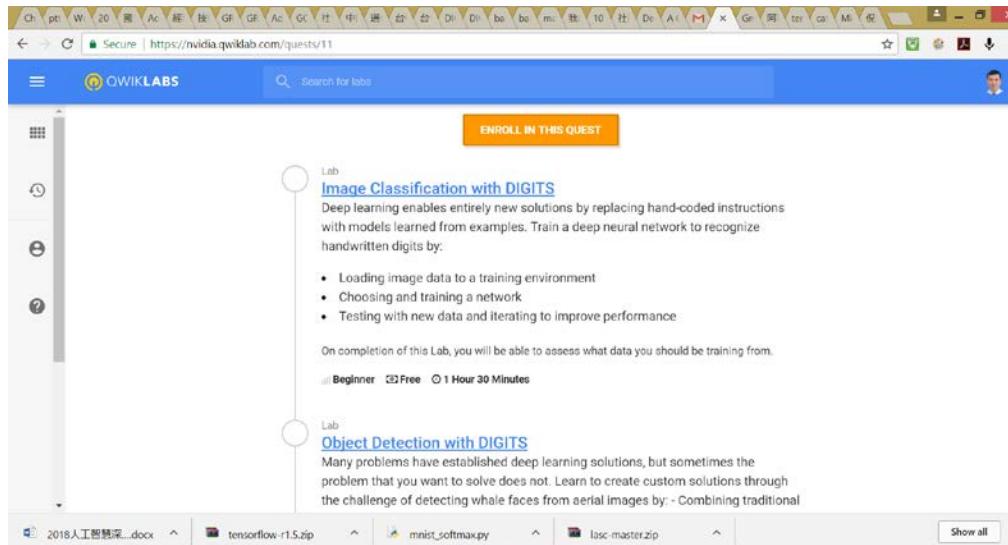


Image Classification with DIGITS

- Nvidia 雲端網站

https://nvidia.qwiklab.com/users/sign_in?locale=en



影像分類

- 影像數據分析理論與案例
 - 1. 運用 Google Tensorflow 建立你的影像分類專案
 - 2. 訓練實驗
 - 3. 預測實驗

Play Neural Network

- Tinker With a Neural Network Right Here in Your Browser.
- Don't Worry, You Can't Break It. We Promise.
- <http://playground.tensorflow.org/>

◀ ▶ C playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle®Dataset=reg-plane&learningRate=0.03®ularizationRate=0&noise=0&... ☆

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

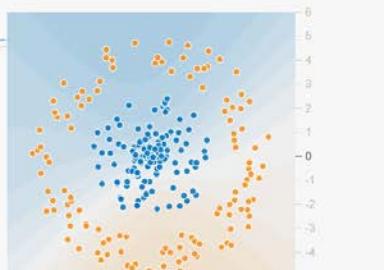
DATA
Which dataset do you want to use?

Ratio of training to test data: 50%
Noise: 0
Data size: 10

FEATURES
Which properties do you want to feed in?
 X_1 X_2 X_1^2 X_2^2

2 HIDDEN LAYERS
+ - 4 neurons + - 2 neurons
The outputs are mixed with varying weights, shown by the thickness of the lines.

OUTPUT
Test loss 0.512
Training loss 0.483



The screenshot shows a web browser displaying the TensorFlow website at https://www.tensorflow.org/get_started/. The page has a bright orange header with the TensorFlow logo and navigation links for Install, Develop (which is underlined), API r1.5, Deploy, Extend, Community, Versions, a search bar, and GitHub. Below the header is a secondary navigation bar with links for GET STARTED, PROGRAMMER'S GUIDE, TUTORIALS, PERFORMANCE, and MOBILE. The main content area is titled "Getting Started". On the left, there's a sidebar with sections for Getting Started (including sub-links for Getting Started, ML Beginners, and TensorFlow), Details (Checkpoints, Feature Columns, Datasets Quick Start, Creating Custom Estimators), and TensorFlow Versions. The main content area starts with a paragraph about TensorFlow being a tool for machine learning, designed for deep neural network models. It then discusses the high-level APIs and lists two recommended documents: "Getting Started for ML Beginners" and "Getting Started with TensorFlow". Finally, it suggests reading "Checkpoints" and "Feature Columns" to understand key features.

Getting Started

TensorFlow is a tool for machine learning. While it contains a wide range of functionality, TensorFlow is mainly designed for deep neural network models.

TensorFlow provides many APIs. This section focuses on the high-level APIs. If you are new to TensorFlow, begin by reading one of the following documents:

- [Getting Started for ML Beginners](#), which is aimed at readers new to machine learning.
- [Getting Started with TensorFlow](#), which is aimed at readers who have experience in machine learning.

Then, read the following documents, which demonstrate the key features in the high-level APIs:

- [Checkpoints](#), which explains how to save training progress and resume where you left off.
- [Feature Columns](#), which shows how an Estimator can handle a variety of input data types without changes to the model.

MNIST For ML Beginners

- <https://www.tensorflow.org/versions/r0.12/tutorials/mnist/beginners/>

The screenshot shows a web browser displaying the TensorFlow r0.12 Guides page. The URL in the address bar is <https://www.tensorflow.org/versions/r0.12/tutorials/mnist/beginners/>. The page has a yellow header with tabs for 'TensorFlow™', 'Current', 'r0.12', 'Getting Started r0.12', 'Guides r0.12' (which is underlined), and 'API r0.12'. There is also a search bar and a GitHub link. The main content area has a blue header 'MNIST For ML Beginners'. Below it, a paragraph explains the tutorial is for beginners and links to a faster paced tutorial. It then describes the tradition of printing 'Hello World' in programming and compares it to machine learning's tradition of using MNIST. A section about the dataset follows, mentioning handwritten digits. At the bottom, there are four handwritten digit images: '5', '0', '4', and '1'. A note at the very bottom says 'It also includes labels for each image, telling us which digit it is. For example, the labels for the above'.

<https://www.tensorflow.org/versions/r0.12/tutorials/mnist/beginners/>

TensorFlow™ Current r0.12 Getting Started r0.12 Guides r0.12 API r0.12 Search GITHUB

Guides r0.12

TUTORIALS HOW TO

MNIST For ML Beginners

This tutorial is intended for readers who are new to both machine learning and TensorFlow. If you already know what MNIST is, and what softmax (multinomial logistic) regression is, you might prefer this [faster paced tutorial](#). Be sure to [install TensorFlow](#) before starting either tutorial.

When one learns how to program, there's a tradition that the first thing you do is print "Hello World." Just like programming has Hello World, machine learning has MNIST.

MNIST is a simple computer vision dataset. It consists of images of handwritten digits like these:

5 0 4 1

It also includes labels for each image, telling us which digit it is. For example, the labels for the above

lasc-master.zip Show all

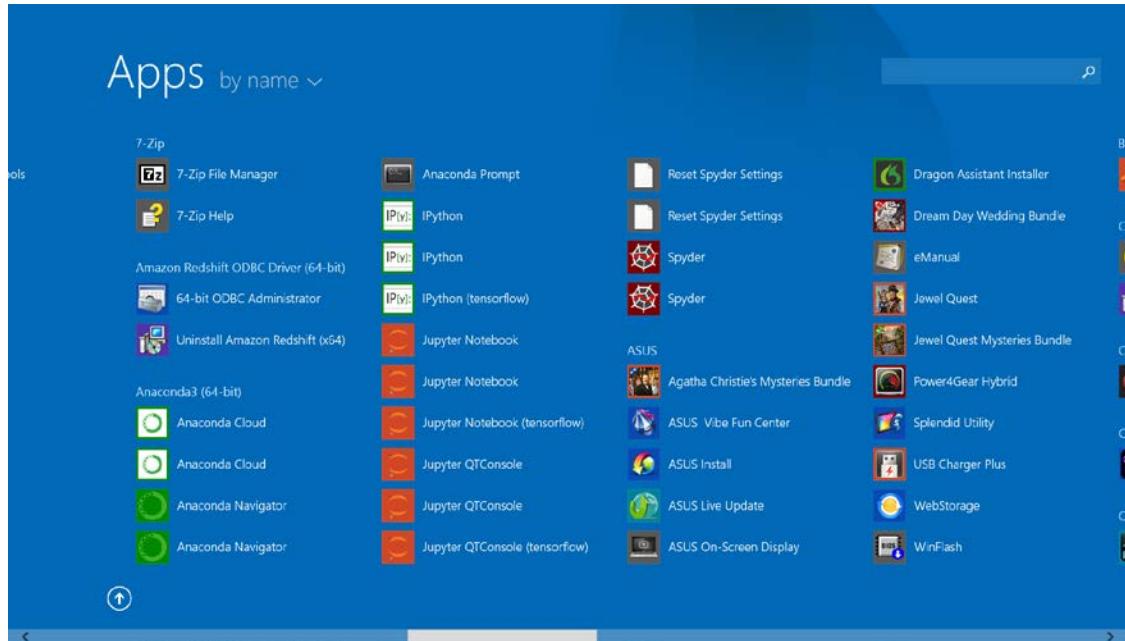
MNIST For ML Beginners

- <https://www.tensorflow.org/versions/r0.12/tutorials/mnist/beginners/>
- Download mnist_softmax.py code(下載檔是html)

The screenshot shows a web browser displaying the TensorFlow r0.12 tutorial page for "MNIST For ML Beginners". The URL in the address bar is <https://www.tensorflow.org/versions/r0.12/tutorials/mnist/beginners/>. The page has a yellow header with the TensorFlow logo and navigation links for Current, r0.12, Getting Started r0.12, Guides r0.12, API r0.12, Search, and GITHUB. The main content area starts with a heading: "give you code to do that later! -- but rather to dip a toe into using TensorFlow. As such, we're going to start with a very simple model, called a Softmax Regression." Below this, there is a sidebar titled "Contents" with links to "About this tutorial", "The MNIST Data", "Softmax Regressions", "Implementing the Regression", "Training", and "Evaluating Our Model". The main content continues with sections on "About this tutorial", "What this tutorial covers", and "What we will accomplish in this tutorial". At the bottom, there is a file download section showing "mnist_softmax.py" and "tasc-master.zip".

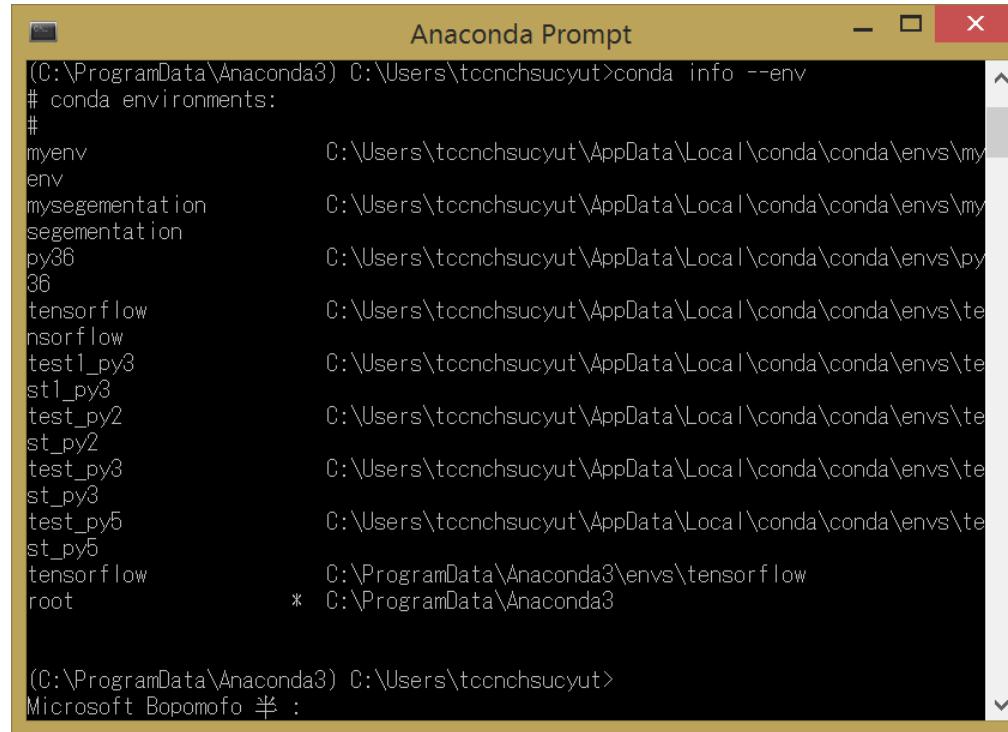
Anaconda Proimppt

- Anaconda Prompt



Anaconda Proimppt

- conda info --env



A screenshot of the Anaconda Prompt window titled "Anaconda Prompt". The command entered is "conda info --env". The output lists various conda environments and their paths:

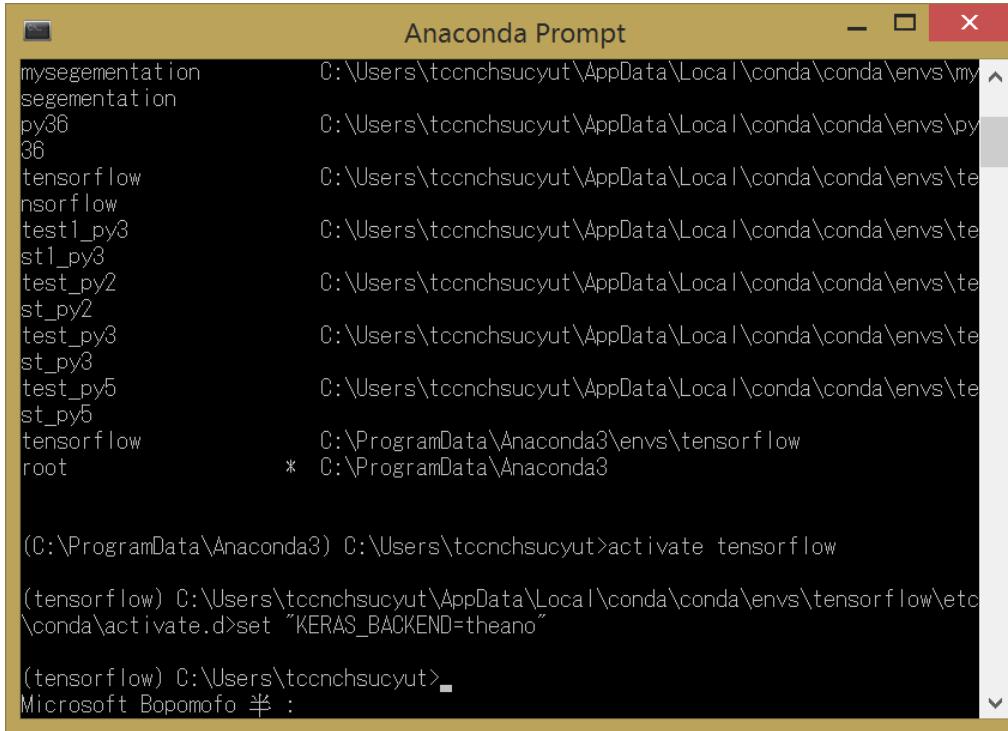
Environment	Path
#	
myenv	C:\Users\tccnchsucyut\AppData\Local\conda\envs\myenv
env	C:\Users\tccnchsucyut\AppData\Local\conda\envs\env
mysegmentation	C:\Users\tccnchsucyut\AppData\Local\conda\envs\mysegmentation
segementation	C:\Users\tccnchsucyut\AppData\Local\conda\envs\segementation
py36	C:\Users\tccnchsucyut\AppData\Local\conda\envs\py36
36	C:\Users\tccnchsucyut\AppData\Local\conda\envs\36
tensorflow	C:\Users\tccnchsucyut\AppData\Local\conda\envs\tensorflow
nsorflow	C:\Users\tccnchsucyut\AppData\Local\conda\envs\nsorflow
test1_py3	C:\Users\tccnchsucyut\AppData\Local\conda\envs\test1_py3
st1_py3	C:\Users\tccnchsucyut\AppData\Local\conda\envs\st1_py3
test_py2	C:\Users\tccnchsucyut\AppData\Local\conda\envs\test_py2
st_py2	C:\Users\tccnchsucyut\AppData\Local\conda\envs\st_py2
test_py3	C:\Users\tccnchsucyut\AppData\Local\conda\envs\test_py3
st_py3	C:\Users\tccnchsucyut\AppData\Local\conda\envs\st_py3
test_py5	C:\Users\tccnchsucyut\AppData\Local\conda\envs\test_py5
st_py5	C:\Users\tccnchsucyut\AppData\Local\conda\envs\st_py5
tensorflow	C:\ProgramData\Anaconda3\envs\tensorflow
root	* C:\ProgramData\Anaconda3

(C:\ProgramData\Anaconda3) C:\Users\tccnchsucyut>

Microsoft Bopomofo 半 :

Anaconda Proimppt

- activate tensorflow



The screenshot shows the Anaconda Prompt window with a title bar "Anaconda Prompt". The left pane lists available environments:

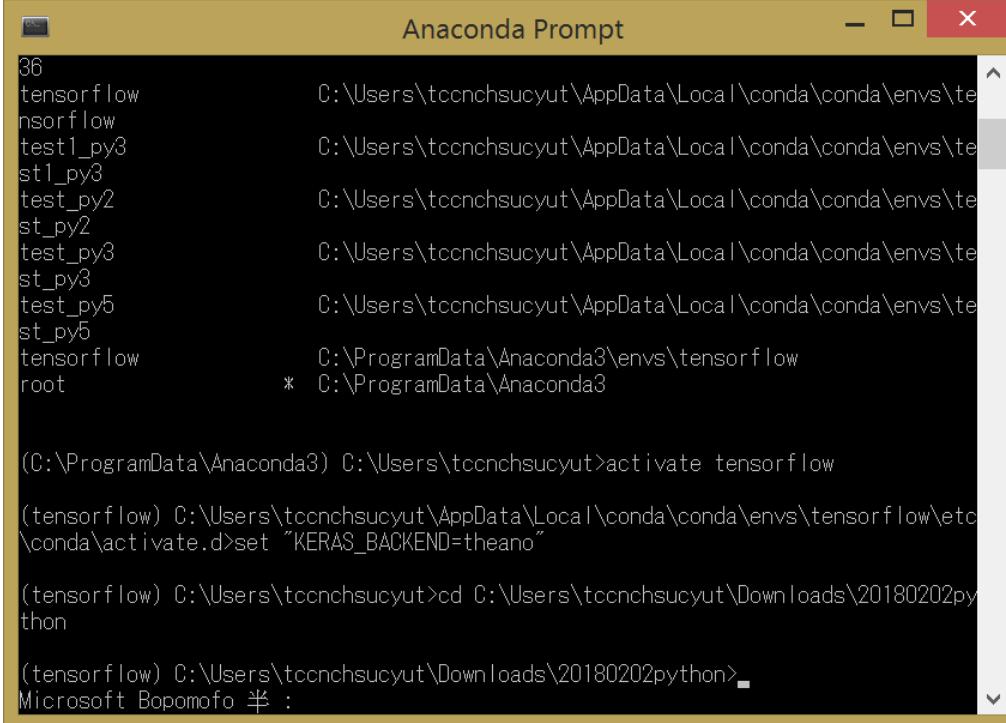
Environment	Path
mysegementation	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\mysegementation
segementation	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\segementation
py36	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\py36
36	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\36
tensorflow	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\tensorflow
nsorflow	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\nsorflow
test1_py3	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\test1_py3
st1_py3	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\st1_py3
test_py2	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\test_py2
st_py2	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\st_py2
test_py3	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\test_py3
st_py3	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\st_py3
test_py5	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\test_py5
st_py5	C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\st_py5
tensorflow	C:\ProgramData\Anaconda3\envs\tensorflow
root	* C:\ProgramData\Anaconda3

The right pane shows the command history and environment activation:

```
(C:\ProgramData\Anaconda3) C:\Users\tccnchhsucyut>activate tensorflow
(tensorflow) C:\Users\tccnchhsucyut\AppData\Local\conda\conda\envs\tensorflow\etc
\conda\activate.d>set "KERAS_BACKEND=theano"
(tensorflow) C:\Users\tccnchhsucyut>
Microsoft Bopomofo 半 :
```

Anaconda Proimppt

- cd C:\Users\tccnchsucyut\Downloads\20180202python



The screenshot shows the Anaconda Prompt window with the title "Anaconda Prompt". The command history at the top lists several environments:

- 36 tensorflow
- nsorflow
- test1_py3
- st1_py3
- test_py2
- st_py2
- test_py3
- st_py3
- test_py5
- st_py5
- tensorflow
- root

The current prompt shows the user has activated the "tensorflow" environment:

```
(C:\ProgramData\Anaconda3) C:\Users\tccnchsucyut>activate tensorflow
```

After activation, the prompt changes to reflect the environment:

```
(tensorflow) C:\Users\tccnchsucyut\AppData\Local\conda\conda\envs\tensorflow\etc\conda\activate.d>set "KERAS_BACKEND=theano"
```

The user then changes the directory to the specified location:

```
(tensorflow) C:\Users\tccnchsucyut>cd C:\Users\tccnchsucyut\Downloads\20180202python
```

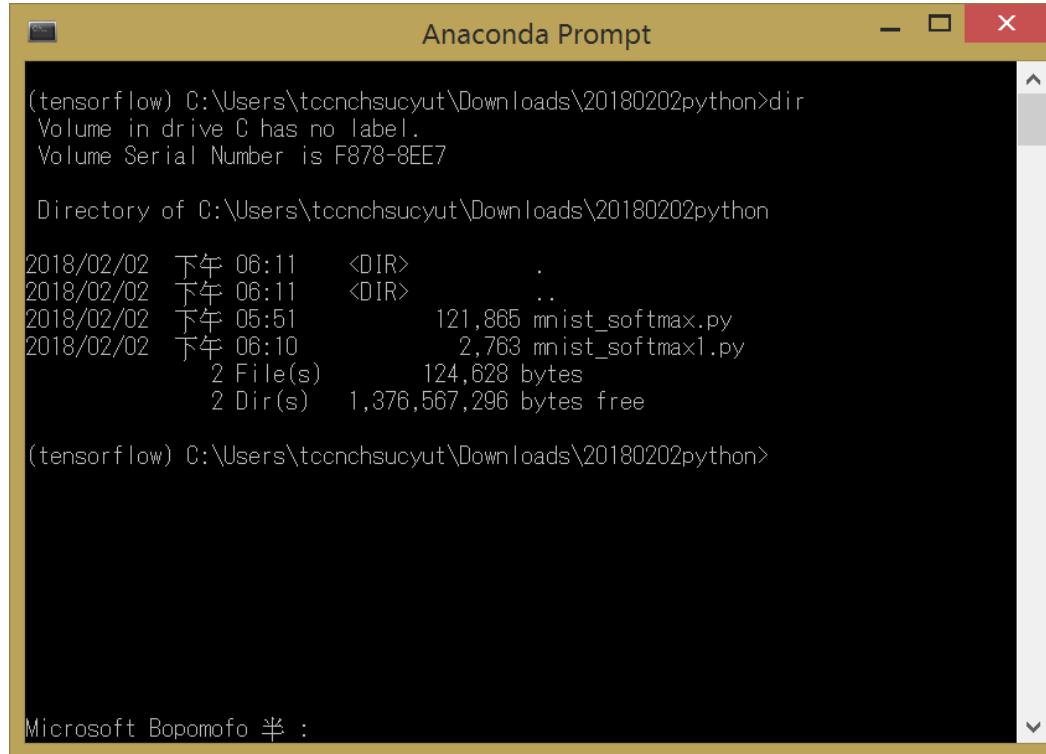
Finally, the user runs a command from the new directory:

```
(tensorflow) C:\Users\tccnchsucyut\Downloads\20180202python>
```

The bottom status bar indicates "Microsoft Bopomofo 半".

Anaconda Proimppt

- dir



The screenshot shows a Windows-style terminal window titled "Anaconda Prompt". The command "(tensorflow) C:\Users\tccnchhsucyut\Downloads\20180202python>dir" is run, displaying the contents of the directory. The output includes the volume information, the directory listing with file names, sizes, and modification dates, and the total disk usage.

```
(tensorflow) C:\Users\tccnchhsucyut\Downloads\20180202python>dir
Volume in drive C has no label.
Volume Serial Number is F878-8EE7

Directory of C:\Users\tccnchhsucyut\Downloads\20180202python

2018/02/02 下午 06:11    <DIR>    .
2018/02/02 下午 06:11    <DIR>    ..
2018/02/02 下午 05:51           121,865  mnist_softmax.py
2018/02/02 下午 06:10           2,763  mnist_softmax1.py
                           2 File(s)   124,628 bytes
                           2 Dir(s)  1,376,567,296 bytes free

(tensorflow) C:\Users\tccnchhsucyut\Downloads\20180202python>
```

Microsoft Bopomofo 半 :

Anaconda Proimppt

- python mnist_softmax1.py (下載檔是mnist_softmax 是html)

The screenshot shows a Windows-style terminal window titled "Anaconda Prompt". The command `python mnist_softmax1.py` is run, followed by a series of log messages indicating the download and extraction of TensorFlow datasets. The CPU feature detection message at the end indicates AVX and AVX2 support.

```
2018/02/02 下午 06:11 <DIR> .
2018/02/02 下午 06:11 <DIR> ..
2018/02/02 下午 05:51 121,865 mnist_softmax.py
2018/02/02 下午 06:10 2,763 mnist_softmax1.py
2 File(s) 124,628 bytes
2 Dir(s) 1,376,567,296 bytes free

(tensorflow) C:\Users\tccnchsucyut\Downloads\20180202python>python mnist_softmax1.py
Successfully downloaded train-images-idx3-ubyte.gz 9912422 bytes.
Extracting /tmp/tensorflow/mnist/input_data\train-images-idx3-ubyte.gz
Successfully downloaded train-labels-idx1-ubyte.gz 28881 bytes.
Extracting /tmp/tensorflow/mnist/input_data\train-labels-idx1-ubyte.gz
Successfully downloaded t10k-images-idx3-ubyte.gz 1648877 bytes.
Extracting /tmp/tensorflow/mnist/input_data\t10k-images-idx3-ubyte.gz
Successfully downloaded t10k-labels-idx1-ubyte.gz 4542 bytes.
Extracting /tmp/tensorflow/mnist/input_data\t10k-labels-idx1-ubyte.gz
2018-02-02 18:14:19.590134: I C:\tf_jenkins\home\workspace\rel-win\M\windows\PY\35\tensorflow\core\platform\cpu_feature_guard.cc:137] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX AVX2
0.9181

(tensorflow) C:\Users\tccnchsucyut\Downloads\20180202python>
(tensorflow) C:\Users\tccnchsucyut\Downloads\20180202python>
Microsoft Bopomofo 半 :
```

handwritten digits

- a large number of handwritten digits, known as training examples.



Convolutional Neural Network

- A Guide to TF Layers: Building a Convolutional Neural Network

The screenshot shows a web browser displaying the TensorFlow website at <https://www.tensorflow.org/tutorials/layers>. The page title is "TensorFlow™ TUTORIALS". The main content area is titled "A Guide to TF Layers: Building a Convolutional Neural Network". It describes the TensorFlow layers module and its API, mentioning how it facilitates the creation of dense and convolutional layers, activation functions, and dropout regularization. It also notes that the tutorial will teach how to build a CNN to recognize handwritten digits from the MNIST dataset. Below this text, there is a grid of nine handwritten digits (0-9) in a 3x3 layout. To the right of the main content, a sidebar lists various sections of the tutorial, including "Contents", "Getting Started", "Intro to Convolutional Neural Networks", "Building the CNN MNIST Classifier", "Input Layer", "Convolutional Layer #1", "Pooling Layer #1", "Convolutional Layer #2 and Pooling Layer #2", "Dense Layer", "Logits Layer", "Generate Predictions", "Calculate Loss", "Configure the Training Op", and "Add evaluation". At the bottom of the page, there is a code snippet for creating a TensorFlow program:

```
from __future__ import absolute_import  
from __future__ import division
```

A Guide to TF Layers: Building a Convolutional Neural Network

- <https://www.tensorflow.org/tutorials/layers>

The screenshot shows a web browser displaying the TensorFlow website at <https://www.tensorflow.org/tutorials/layers>. The page has a yellow header with the word "Develop" underlined. The main content area features the title "A Guide to TF Layers: Building a Convolutional Neural Network". Below the title, there is a paragraph of text explaining the purpose of the tutorial, followed by a grid of handwritten digits from 0 to 9. A sidebar on the left lists various TensorFlow tutorials and images, while another sidebar on the right lists the contents of the current tutorial.

Tutorials

- Images
 - [A Guide to TF Layers: Building a Convolutional Neural Network](#)
 - Image Recognition
 - How to Retrain Inception's Final Layer for New Categories
 - Convolutional Neural Networks
- Sequences
 - Recurrent Neural Networks
 - Neural Machine Translation (seq2seq) Tutorial
 - Recurrent Neural Networks for

A Guide to TF Layers: Building a Convolutional Neural Network

The TensorFlow `layers` module provides a high-level API that makes it easy to construct a neural network. It provides methods that facilitate the creation of dense (fully connected) layers and convolutional layers, adding activation functions, and applying dropout regularization. In this tutorial, you'll learn how to use `layers` to build a convolutional neural network model to recognize the handwritten digits in the MNIST data set.

The MNIST dataset comprises 60,000 training examples and 10,000 test examples of the handwritten digits 0–9, formatted as 28x28-pixel monochrome images.

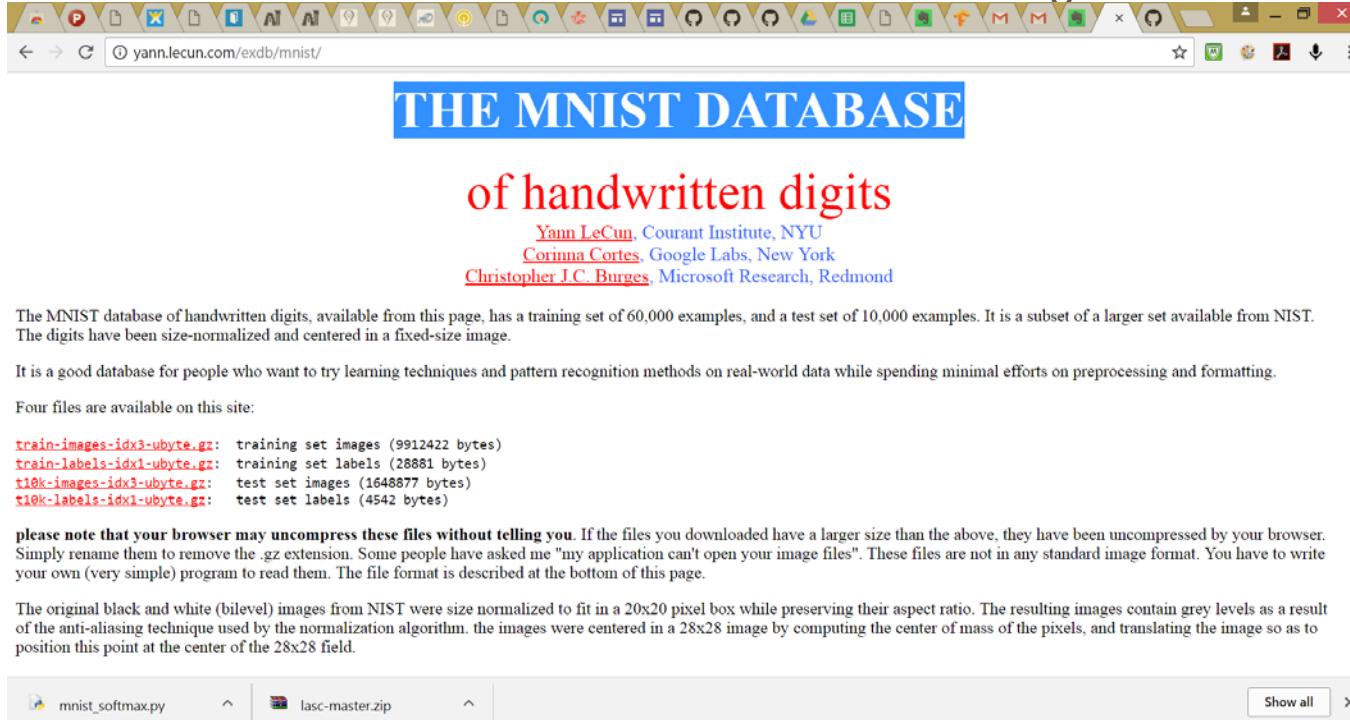
Contents

- Getting Started
- Intro to Convolutional Neural Networks
- Building the CNN MNIST Classifier
- Input Layer
- Convolutional Layer #1
- Pooling Layer #1
- Convolutional Layer #2 and Pooling Layer #2
- Dense Layer
- Logits Layer
- Generate Predictions

tensorflow-r1.5.zip mnist_softmax.py lasc-master.zip Show all

THE MNIST DATABASE

- THE MNIST DATABASE of handwritten digits



The screenshot shows a web browser window with the URL yann.lecun.com/exdb/mnist/ in the address bar. The page title is "THE MNIST DATABASE". Below the title, the text "of handwritten digits" is displayed in large red letters. Underneath, three names are listed with their respective institutions: "Yann LeCun, Courant Institute, NYU", "Corinna Cortes, Google Labs, New York", and "Christopher J.C. Burges, Microsoft Research, Redmond". A descriptive paragraph follows, stating: "The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image." Another paragraph explains: "It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting." Below this, it says: "Four files are available on this site:" followed by a list of four files: "train-images-idx3-ubyte.gz", "train-labels-idx1-ubyte.gz", "t10k-images-idx3-ubyte.gz", and "t10k-labels-idx1-ubyte.gz". At the bottom, there is a note: "please note that your browser may uncompress these files without telling you. If the files you downloaded have a larger size than the above, they have been uncompressed by your browser. Simply rename them to remove the .gz extension. Some people have asked me "my application can't open your image files". These files are not in any standard image format. You have to write your own (very simple) program to read them. The file format is described at the bottom of this page." At the very bottom of the browser window, there are two tabs: "mnist_softmax.py" and "lasc-master.zip".

The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting.

Four files are available on this site:

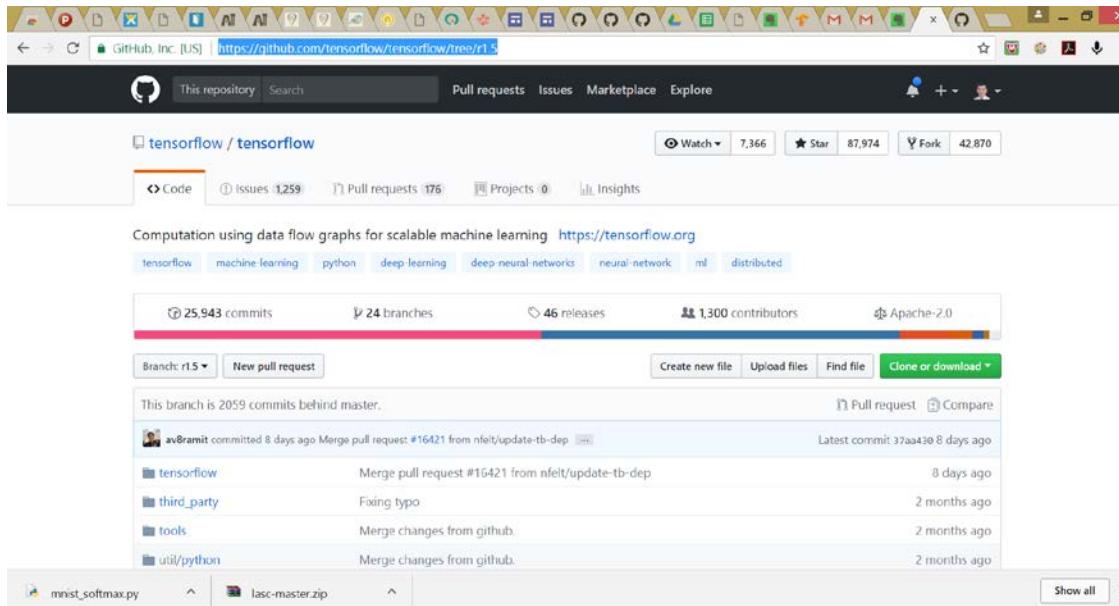
[train-images-idx3-ubyte.gz](#): training set images (9912422 bytes)
[train-labels-idx1-ubyte.gz](#): training set labels (28881 bytes)
[t10k-images-idx3-ubyte.gz](#): test set images (1648877 bytes)
[t10k-labels-idx1-ubyte.gz](#): test set labels (4542 bytes)

please note that your browser may uncompress these files without telling you. If the files you downloaded have a larger size than the above, they have been uncompressed by your browser. Simply rename them to remove the .gz extension. Some people have asked me "my application can't open your image files". These files are not in any standard image format. You have to write your own (very simple) program to read them. The file format is described at the bottom of this page.

The original black and white (bilevel) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field.

THE MNIST DATABASE

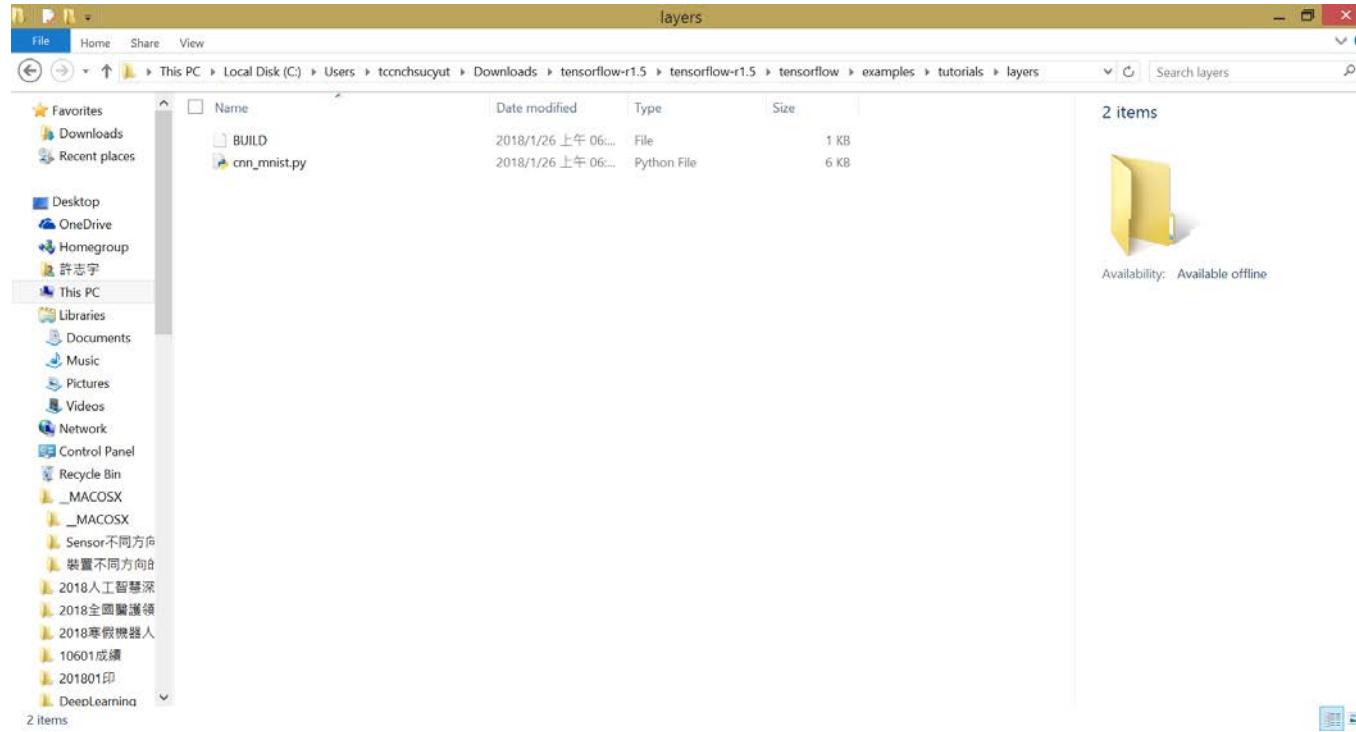
- tensorflow/tensorflow
- <https://github.com/tensorflow/tensorflow/tree/r1.5>



Anaconda Proimppt



clone->tensorflow-r5.zip
Unzip
C:\Users\tccnchsucut\Downloads\tensorflow-r1.5\tensorflow-r1.5\tensorflow\examples\tutorials\layers



Anaconda Proimppt



clone->tensorflow-r5.zip
Unzip
cd C:\Users\tccnchhsucyut\Downloads\tensorflow-r1.5\tensorflow-r1.5\tensorflow\examples\tutorials\layers

A screenshot of the Anaconda Prompt window titled "Anaconda Prompt". The window has a yellow header bar and a black body. It displays two lines of command history:

```
(tensorflow) C:\Users\tccnchhsucyut\Downloads\20180202python>cd C:\Users\tccnchhsucyut\Downloads\tensorflow-r1.5\tensorflow-r1.5\tensorflow\examples\tutorials\layers  
(tensorflow) C:\Users\tccnchhsucyut\Downloads\tensorflow-r1.5\tensorflow-r1.5\tensorflow\examples\tutorials\layers>
```

The bottom of the window shows the text "Microsoft Bopomofo 半 :" in a light blue font.

Anaconda Proimppt

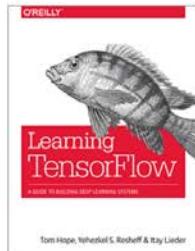
- python cnn_mnist.py

```
0.10290141 0.10225116 0.10297754 0.10274982]
[ 0.10132676 0.09075262 0.11322413 0.10949109 0.10700565 0.09303672
 0.0952975 0.08676055 0.09630577 0.10679925]
[ 0.10058299 0.09638202 0.09403892 0.10394502 0.09299175 0.09895375
 0.10612112 0.11052606 0.09641723 0.10004117]
[ 0.0918543 0.09487323 0.11517952 0.09978802 0.09876621 0.09776193
 0.10310867 0.09516761 0.09996022 0.10354028]
[ 0.10540446 0.0920971 0.10435287 0.11700255 0.09270071 0.09578692
 0.08895756 0.10383614 0.09363227 0.10622943]
[ 0.08452667 0.10153443 0.098043 0.1070532 0.1077362 0.09853619
 0.10474801 0.09610903 0.09717435 0.10453887]
[ 0.09537908 0.10196666 0.085955 0.10769309 0.10078052 0.09449783
 0.09239599 0.11553453 0.09735549 0.10844189]
[ 0.10209614 0.09574585 0.10281904 0.11007897 0.09921789 0.09551911
 0.0918666 0.10614 0.09172029 0.10479616]
[ 0.08489551 0.10104841 0.10988043 0.09473744 0.11148597 0.09699973
 0.09639002 0.1015711 0.09439988 0.1085914 ]
[ 0.09592968 0.09967948 0.10436896 0.10166489 0.09499774 0.0932716
 0.1000098 0.099464 0.09587926 0.11473463]
[ 0.09183238 0.09960435 0.10610702 0.0989664 0.10510809 0.09951506
 0.09882346 0.09686585 0.0957728 0.10740463]
[ 0.0947644 0.10636142 0.09538183 0.11380132 0.09803268 0.09493954
 0.10212552 0.10385285 0.09145048 0.09928995]] (64.072 sec)

Microsoft Bopomofo 半 :
```

Learning TensorFlow A Guide to Building Deep Learning Systems

- <http://shop.oreilly.com/product/0636920063698.do>



Learning TensorFlow

A Guide to Building Deep Learning Systems

By [Itay Lieder](#), [Yehezkel Resheff](#), [Tom Hope](#)

Publisher: [O'Reilly Media](#)

Release Date: August 2017

Pages: 242

Roughly inspired by the human brain, deep neural networks trained with large amounts of data can solve complex tasks with unprecedented accuracy. This practical book provides an end-to-end guide to TensorFlow, the leading open source software library that helps you build and train neural networks for computer vision, natural language processing (NLP), speech recognition, and general predictive analytics.

Authors Tom Hope, Yehezkel Resheff, and Itay Lieder provide a hands-on approach to TensorFlow fundamentals for a broad technical audience—from data scientists and engineers to students and researchers. You'll begin by working through some basic examples in TensorFlow before diving deeper into topics such as neural network architectures, TensorBoard visualization, TensorFlow abstraction libraries, and multithreaded input pipelines. Once you finish this book, you'll know how to build and deploy production-ready deep learning systems in

[Read on Safari
with a 10-day trial](#)

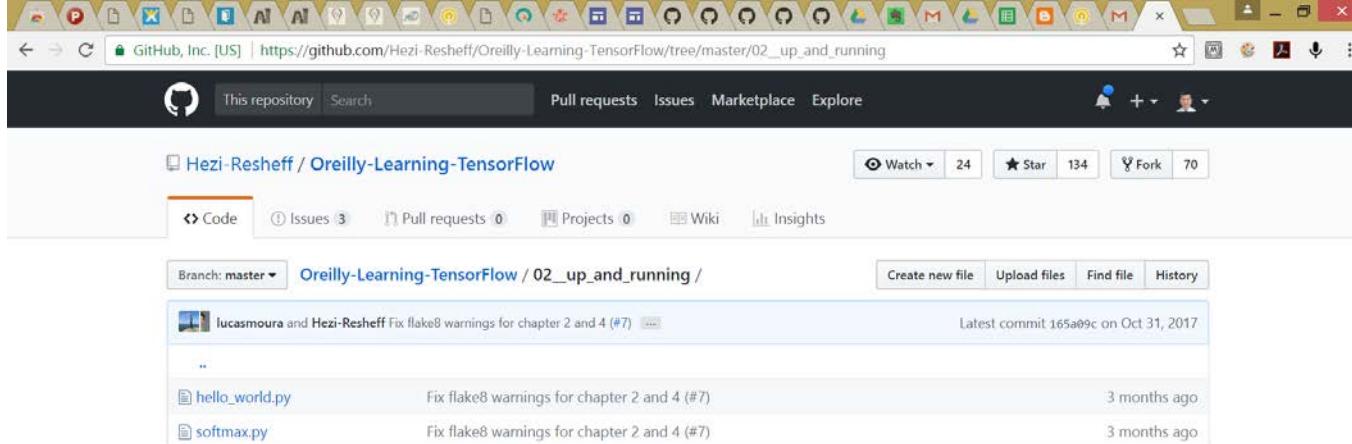
[Start your free trial](#)

[Buy on Amazon](#)

Where's the cart? Now you can get everything on [Safari](#). To purchase books, visit [Amazon](#) or your favorite retailer.
Questions? [See our FAQ](#) or contact customer service:
1-800-889-8969 / 707-827-7019

Hezi-Resheff/Oreilly-Learning-TensorFlow

- <https://github.com/Hezi-Resheff/Oreilly-Learning-TensorFlow>



Hezi-Resheff/Oreilly-Learning-TensorFlow

- https://github.com/Hezi-Resheff/Oreilly-Learning-TensorFlow/tree/master/02_up_and_running
- 隨堂測驗
- 執行 softmax.py

softmax.py

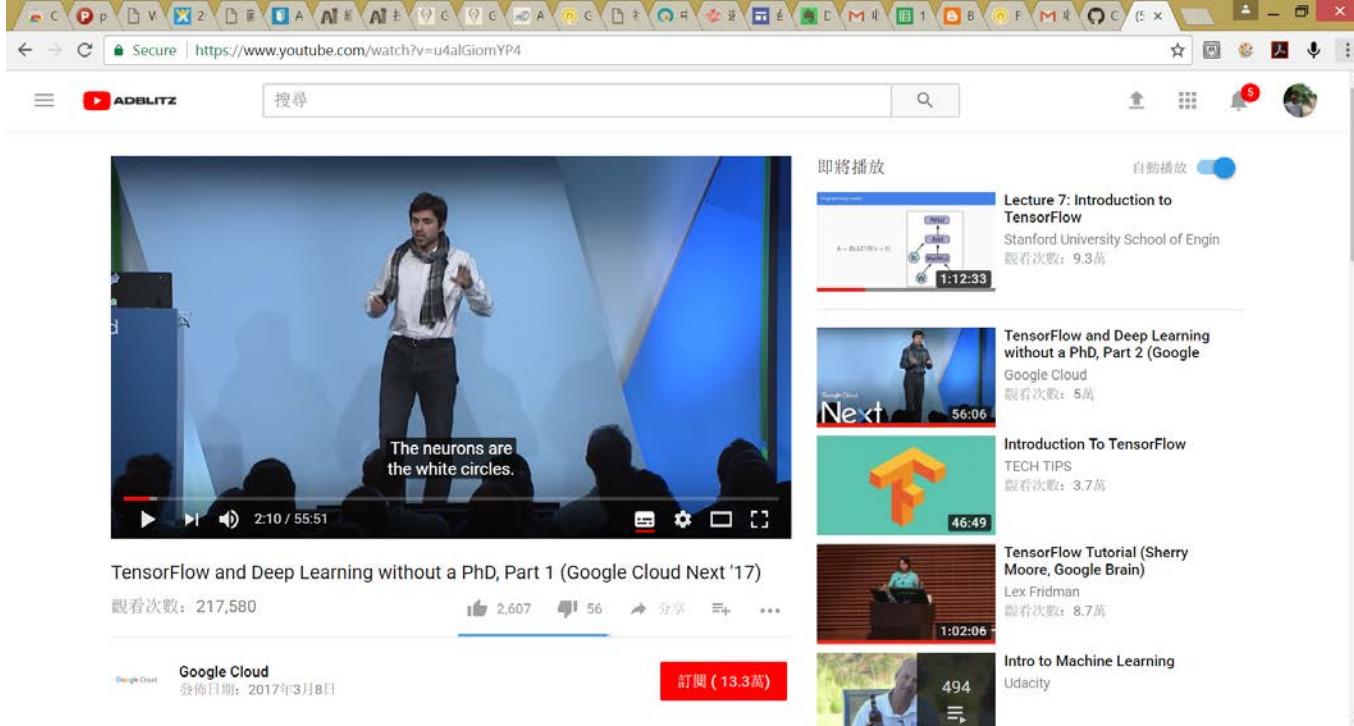
- import tensorflow as tf
- from tensorflow.examples.tutorials.mnist import input_data
- DATA_DIR = '/tmp/data'
- NUM_STEPS = 1000
- MINIBATCH_SIZE = 100
- data = input_data.read_data_sets(DATA_DIR, one_hot=True)
- x = tf.placeholder(tf.float32, [None, 784])
- W = tf.Variable(tf.zeros([784, 10]))
- y_true = tf.placeholder(tf.float32, [None, 10])
- y_pred = tf.matmul(x, W)
- cross_entropy = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(
● logits=y_pred, labels=y_true))
- gd_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)
- correct_mask = tf.equal(tf.argmax(y_pred, 1), tf.argmax(y_true, 1))
- accuracy = tf.reduce_mean(tf.cast(correct_mask, tf.float32))

softmax.py

- with tf.Session() as sess:
- # Train
- sess.run(tf.global_variables_initializer())
- for _ in range(NUM_STEPS):
- batch_xs, batch_ys =
 data.train.next_batch(MINIBATCH_SIZE)
- sess.run(gd_step, feed_dict={x: batch_xs, y_true: batch_ys})
- # Test
- ans = sess.run(accuracy, feed_dict={x: data.test.images,
 y_true: data.test.labels})
- print("Accuracy: {:.4}%".format(ans*100))

softmax.py

- Videos <https://youtu.be/u4alGiomYP4>





The study plan

- Short-Term Project Report Presentation



Major Course Assignments

- In-Class activity, assignment done in class
- Individual assignment at home
- Short-Term Project in teams

Major Course Assignments

Individual Assignment Activity of Unit 3:

Run the Lab: Image Classification with DIGITS

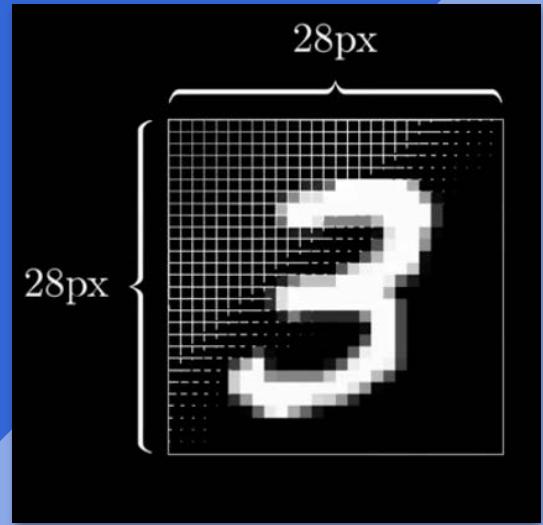
Individual Assignment Activity of Unit 5: (option)

Run the Lab: Medical Image Segmentation with DIGITS

<https://www.nvidia.com/en-us/deep-learning-ai/education/>



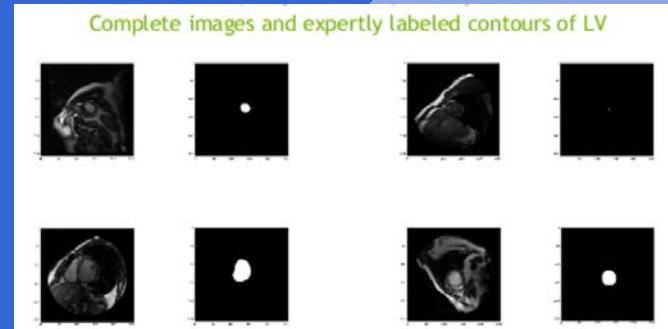
Individual Assignment Activity of Unit 3:



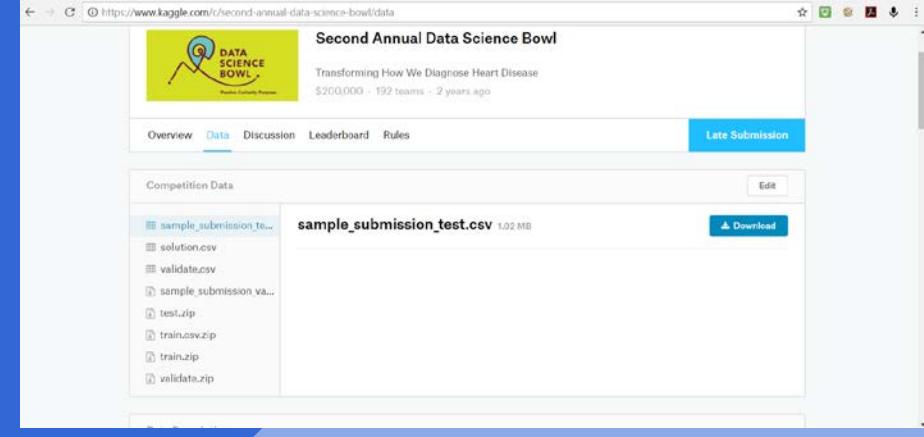
- Learn how to leverage deep neural networks (DNN) within the deep learning workflow to recognize handwritten characters using NVIDIA DIGITS™.

Short-Term Project in teams

- Segment MRI images to measure parts of the heart by setting up a computer vision workflow using deep learning.



Short-Term Project Dataset



- Data Science Bowl Competition
Second Annual Data Science Bowl
- Transforming How We Diagnose Heart Disease
- <https://www.kaggle.com/c/second-annual-data-science-bowl>



Short-Term Project in teams

After completing this project, you will:

- Know how Convolutional Neural Networks (CNNs) work.
- Know how to prepare data for CNNs.
- Know how to design architecture of CNNs.
- Know how to tune CNNs to a problem.
- Know how to save an CNN model and use it to make predictions.



Short-Term Project Report

Report Contents involved as follows:

Section 1: Overview of concepts of Deep Neural Networks

Section 2: Design a CNNs architecture for medical image segmentation.

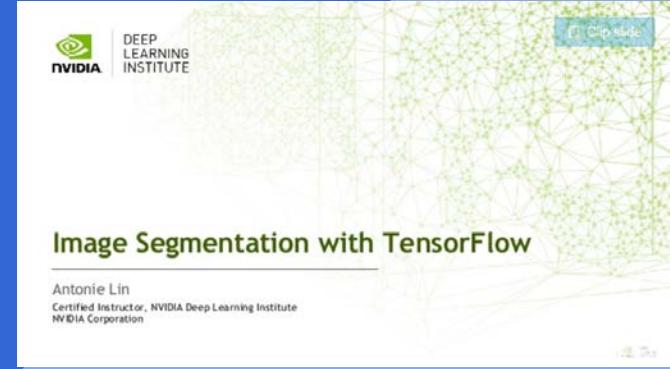
Section 3: Impemetation of the CNNs architecture in Tensorflow and Python.

Section 4: Evaluation of the Design of a CNNs architecture.

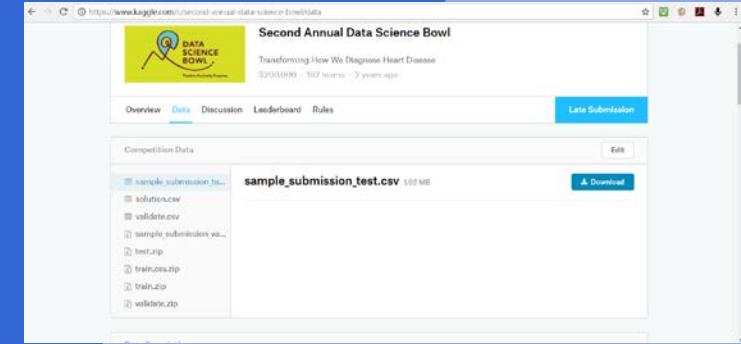


Short-Term Project Resources

- Introduction to Image Segmentation with TensorFlow
- <https://www.slideshare.net/NVIDIATaiwan/nvidia-dli-image-segmentation-with-tensorflow>



Short-Term Project Resources



- Diagnosing Heart Diseases in the Data Science Bowl: 2nd place, Team kunsthart
- <http://blog.kaggle.com/2016/04/13/diagnosing-heart-diseases-with-deep-neural-networks-2nd-place-ira-korshunova/>

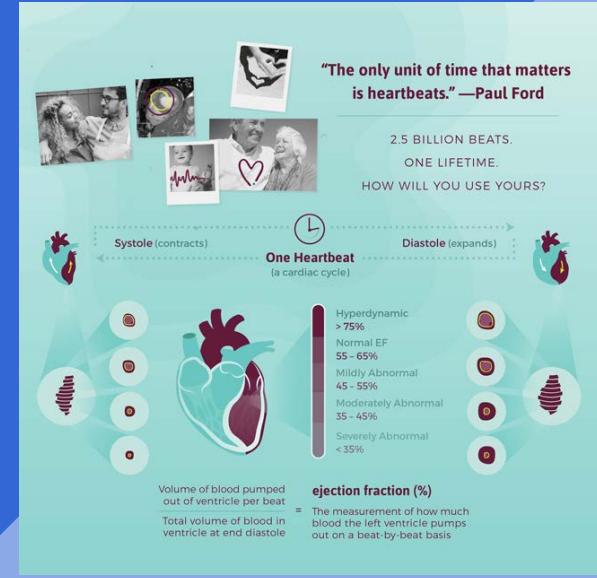
Short-Term Project Resources



- Leading and Winning Team Submissions Analysis
- <https://datasciencebowl.com/leading-and-winning-team-submissions-analysis/>

Short-Term Project Resources

- **Left Atrial Segmentation Challenge 2013**
- <https://github.com/catactg/lasc>





Short-Term Project Resources

- Using DIGITS to train a medical image segmentation network
- [https://github.com/NVIDIA/DIGITS/tree/master/examples/
medical-imaging](https://github.com/NVIDIA/DIGITS/tree/master/examples/medical-imaging)

Let the course make you feel like eating Grapes!

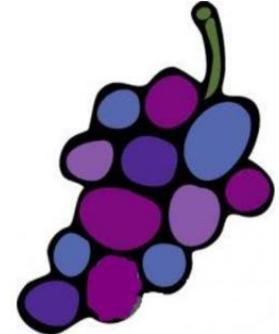
Make you Growth. Make you Recognition.

Make you Achievement. Make you Participation.

Make you Enjoyment.

GRAPE Retention

- Growth
- Recognition
- Achievement
- Participation
- Enjoyment



迴歸分析 (Regression Analysis)

- [TensorFlow Tutorial \(Sherry Moore, Google Brain\)](#)

Apps by name ▾

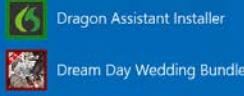
7-Zip



Amazon Redshift ODBC Driver (64-bit)



Anaconda3 (64-bit)



Blender



CDTower



Chrome Apps



CyberLink PowerDri



CyberLink WaveEditor



ASUS



WinFlash

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Applications on root Channels Refresh

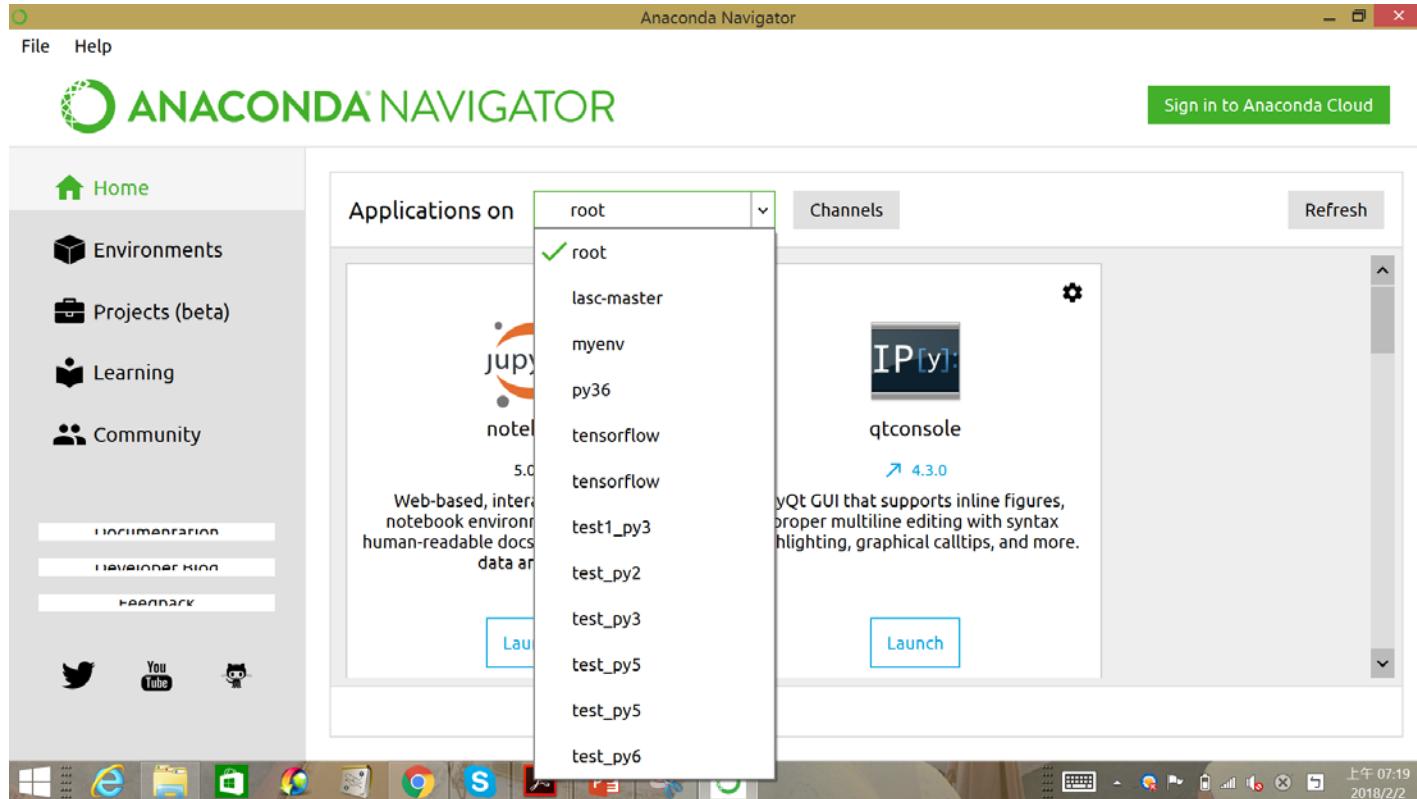
 **jupyter**
notebook
5.0.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
[Launch](#)

 **IP[y]:**
qtconsole
4.3.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
[Launch](#)

Home Environments Projects (beta) Learning Community Documentation Developer Guide Feedback

Setting Environment



Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home Environments Projects (beta) Learning Community Documentation Developer Kit Feedback

Search Environments

Installed Channels Update index... Search

Name	T	Description	Version
alabaster	<input checked="" type="checkbox"/>	Configurable, python 2+3 compatible sphinx theme	0.7.1
anaconda	<input checked="" type="checkbox"/>		4.4.0
anaconda-client	<input checked="" type="checkbox"/>	Anaconda.org command line client library	1.6.0
anaconda-project	<input checked="" type="checkbox"/>	Reproducible, executable project...	0.6.1
asn1crypto	<input checked="" type="checkbox"/>	Asn.1 parser and serializer	0.22.0
astroid	<input checked="" type="checkbox"/>	Abstract syntax tree for python with inference support	1.4.0
astropy	<input checked="" type="checkbox"/>	Community-developed python lib...	1.3.3

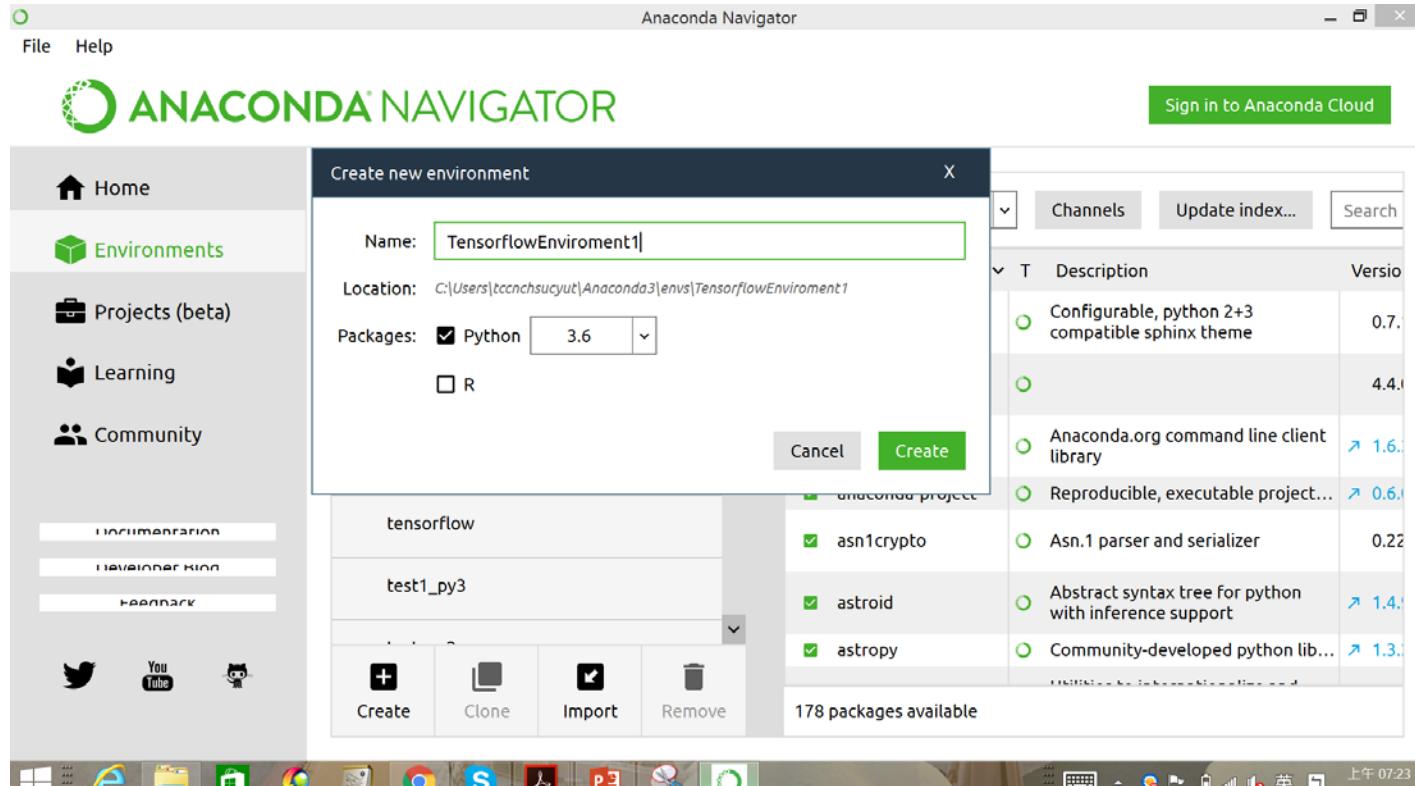
178 packages available

Create Clone Import Remove

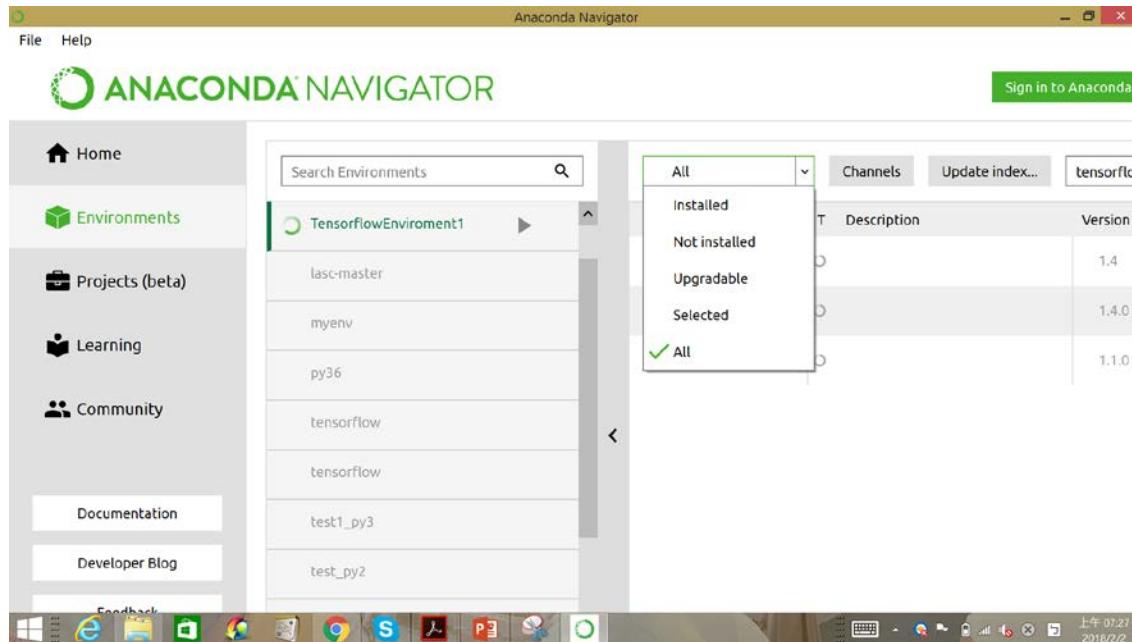
The screenshot shows the Anaconda Navigator application window. On the left is a sidebar with icons for Home, Environments (selected), Projects (beta), Learning, and Community, along with links for Documentation, Developer Kit, and Feedback. Below the sidebar are social media links for Twitter, YouTube, and GitHub. The main area has a search bar for environments and a table of installed packages. The table includes columns for Name, Type (T), Description, and Version. Several packages are listed, including alabaster, anaconda, anaconda-client, anaconda-project, asn1crypto, astroid, and astropy. At the bottom, there are buttons for Create, Clone, Import, and Remove environments.

Enviroment->create

- TensorflowEnviroment1



Search->All->Tensorflow



Tensorflow -> v-> Apply

File Help

Anaconda Navigator



Sign in to Anaconda Cloud

Home

Environments

Projects (beta)

Learning

Community

DOCUMENTATION

DEVELOPER KIT

FEEDBACK

[Twitter](#) [YouTube](#) [GitHub](#)

Install Packages

X

12 packages will be installed

	Name	Unlink	Link	Channel
1	tensorflow	-	1.4.0	conda-forge
2	*bleach	-	1.5.0	conda-forge
3	*html5lib	-	0.9999999	conda-forge
4	*markdown	-	2.6.9	conda-forge
5	*mkl	-	2017.0.3	defaults
6	*numpy	-	1.13.1	defaults
7	*protobuf	-	3.5.1	conda-forge
8	*six	-	1.11.0	conda-forge

* indicates the package is a dependency of a selected package

Cancel Apply

Create Clone Import Remove

3 packages available matching "tensorflow" 1 package selected

Apply

Channels Update index... tensorflow

	Description	Version
1		1.4
2		1.4.0
3		1.1.0



Tensorflow-> v->Apply

Anaconda Navigator

File Help

ANACONDA NAVIGATOR Sign in to Anaconda Cloud

Home Environments Projects (beta) Learning Community Documentation Developer Kit Feedback

Search Environments

All Channels Update index... tensorflow

Name	Description	Version
r-tensorflow	○	1.4
tensorflow	○	1.4.0
tensorflow-gpu	○	1.1.0

TensorflowEnviroment1 ➤

lasc-master

myenv

py36

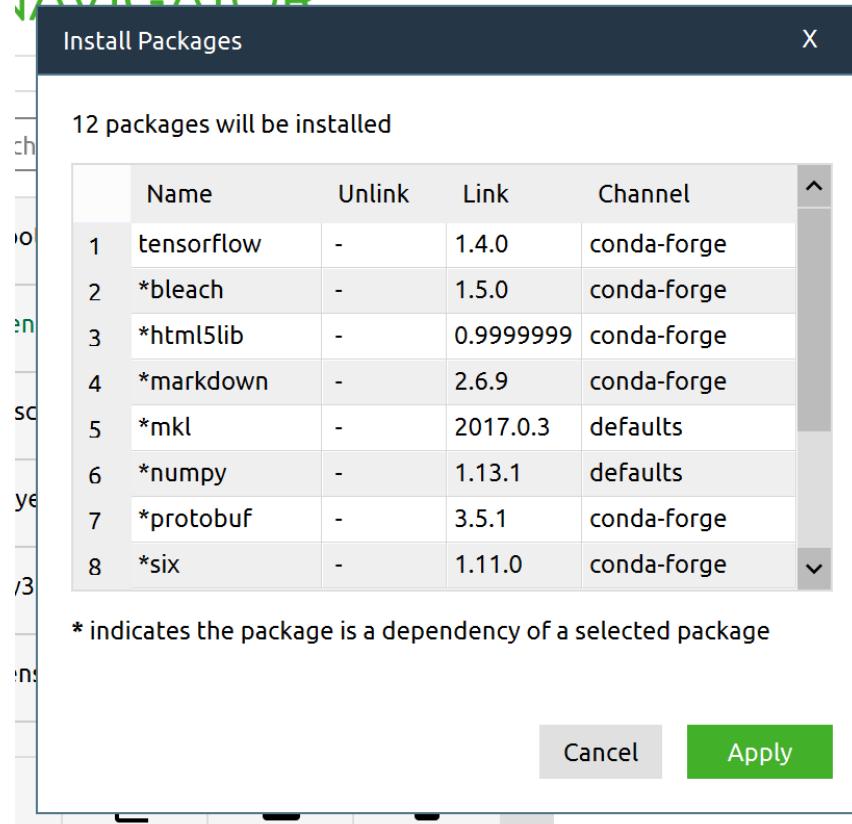
tensorflow

Create Clone Import Remove

3 packages available matching "tensorflow" 1 package selected

The screenshot shows the Anaconda Navigator interface. On the left is a sidebar with links for Home, Environments (which is currently selected), Projects (beta), Learning, and Community, along with links for Documentation, Developer Kit, and Feedback. Below the sidebar are social media icons for Twitter, YouTube, and GitHub. The main area has a search bar at the top. A dropdown menu shows 'All' selected, with 'Channels' and 'Update index...' buttons. A search term 'tensorflow' is entered in the search bar. To the right is a table listing packages: r-tensorflow (version 1.4), tensorflow (version 1.4.0, highlighted in green), and tensorflow-gpu (version 1.1.0). Below the table, it says '3 packages available matching "tensorflow" 1 package selected' and there is a green 'Apply' button. On the far left, a list of environments is shown: root, TensorflowEnviroment1 (selected), lasc-master, myenv, py36, and tensorflow.

Tensorflow-> v-> Apply



Home->Jupyter->Install

Anaconda Navigator

File Help

Sign in to Anaconda Cloud

ANACONDA NAVIGATOR

Applications on TensorflowEnvirom Channels Refresh

Home Environments Projects (beta) Learning Community Documentation Developer Kino Feedback

 Jupyter notebook 5.0.0

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

[Install](#)

 orange3 3.8.0

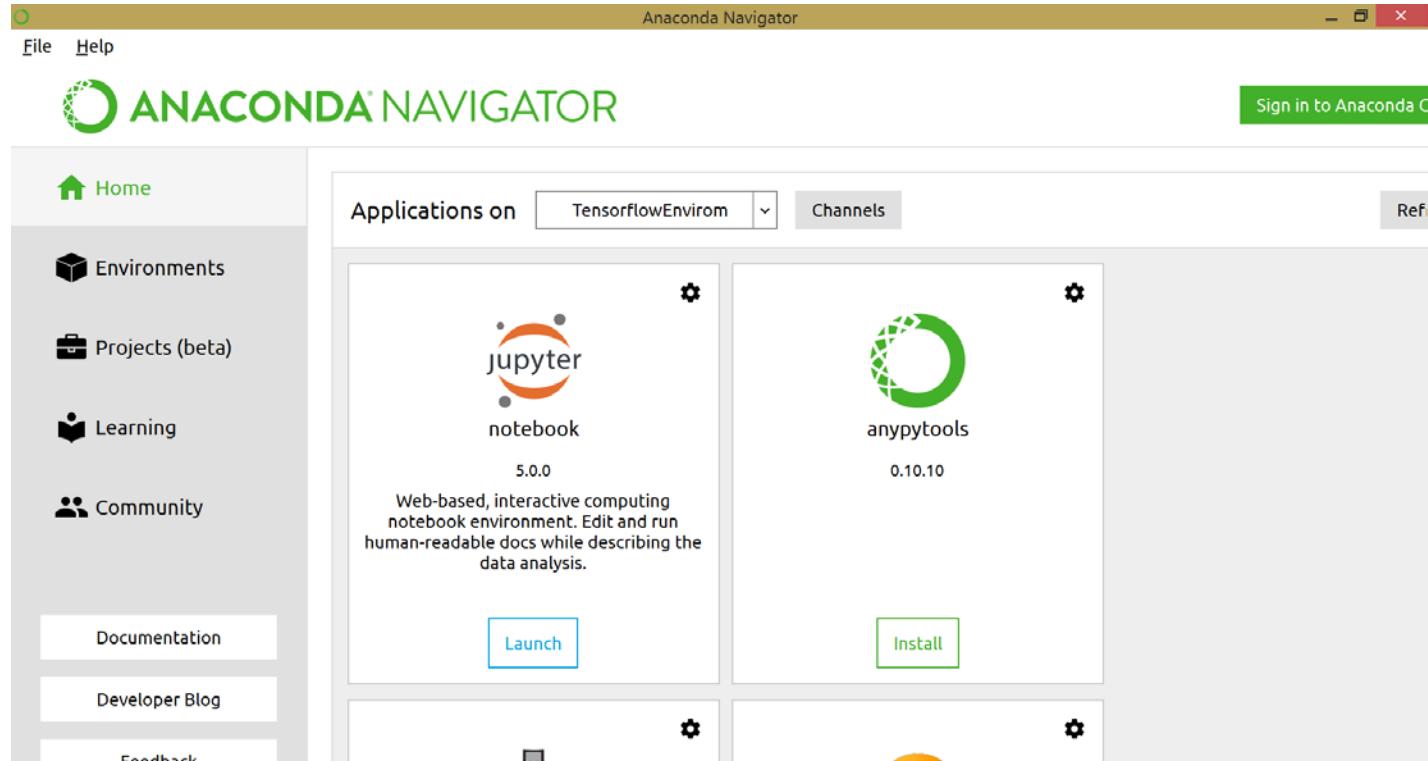
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

[Install](#)

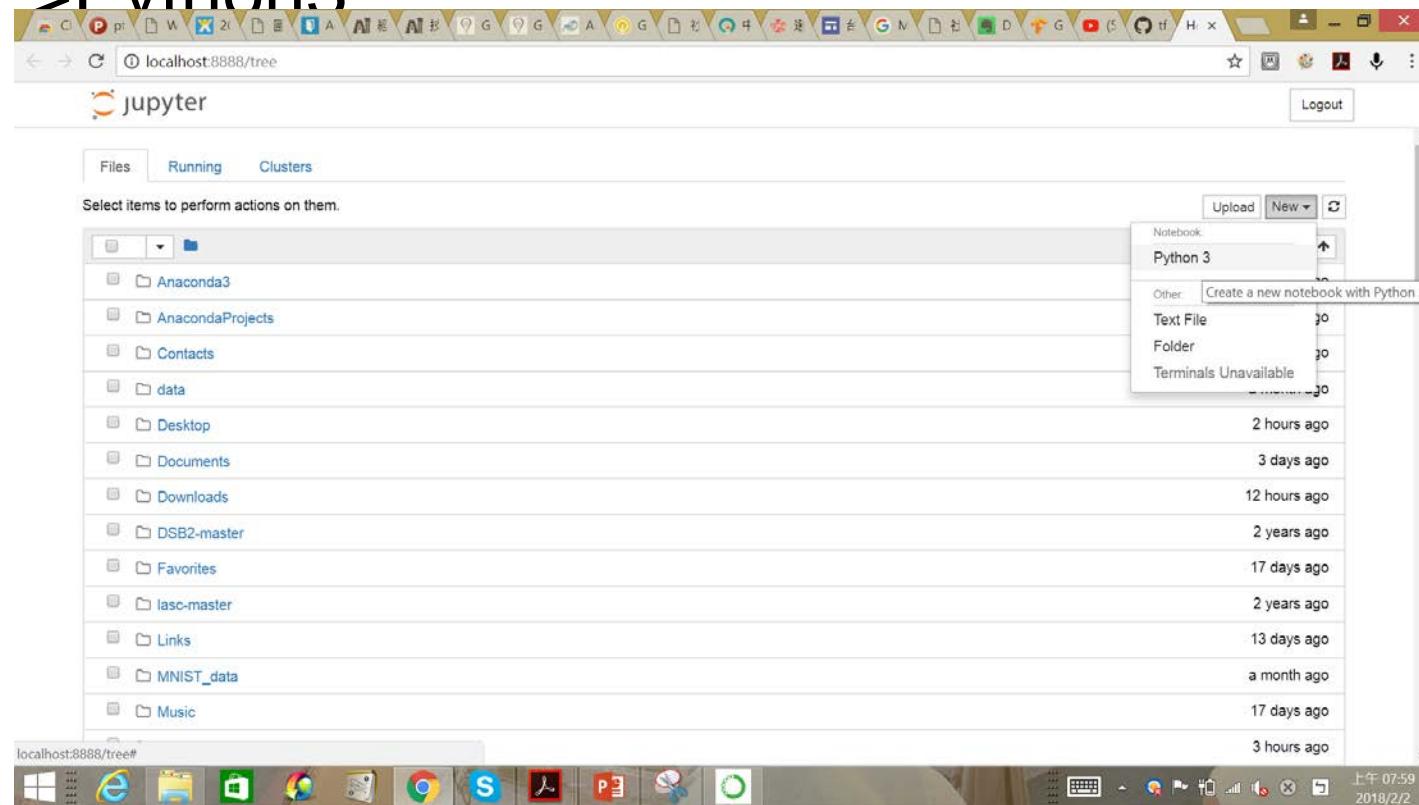
[Twitter](#) [YouTube](#) [GitHub](#)

This screenshot shows the Anaconda Navigator interface. On the left is a sidebar with links for Home, Environments, Projects (beta), Learning, and Community, along with links for Documentation, Developer Kino, and Feedback, and social media icons for Twitter, YouTube, and GitHub. The main area displays two applications: 'Jupyter notebook' (version 5.0.0) and 'orange3' (version 3.8.0). Both applications have an 'Install' button below them. The 'Jupyter notebook' card includes a brief description: 'Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.' The 'orange3' card includes a description: 'Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.'

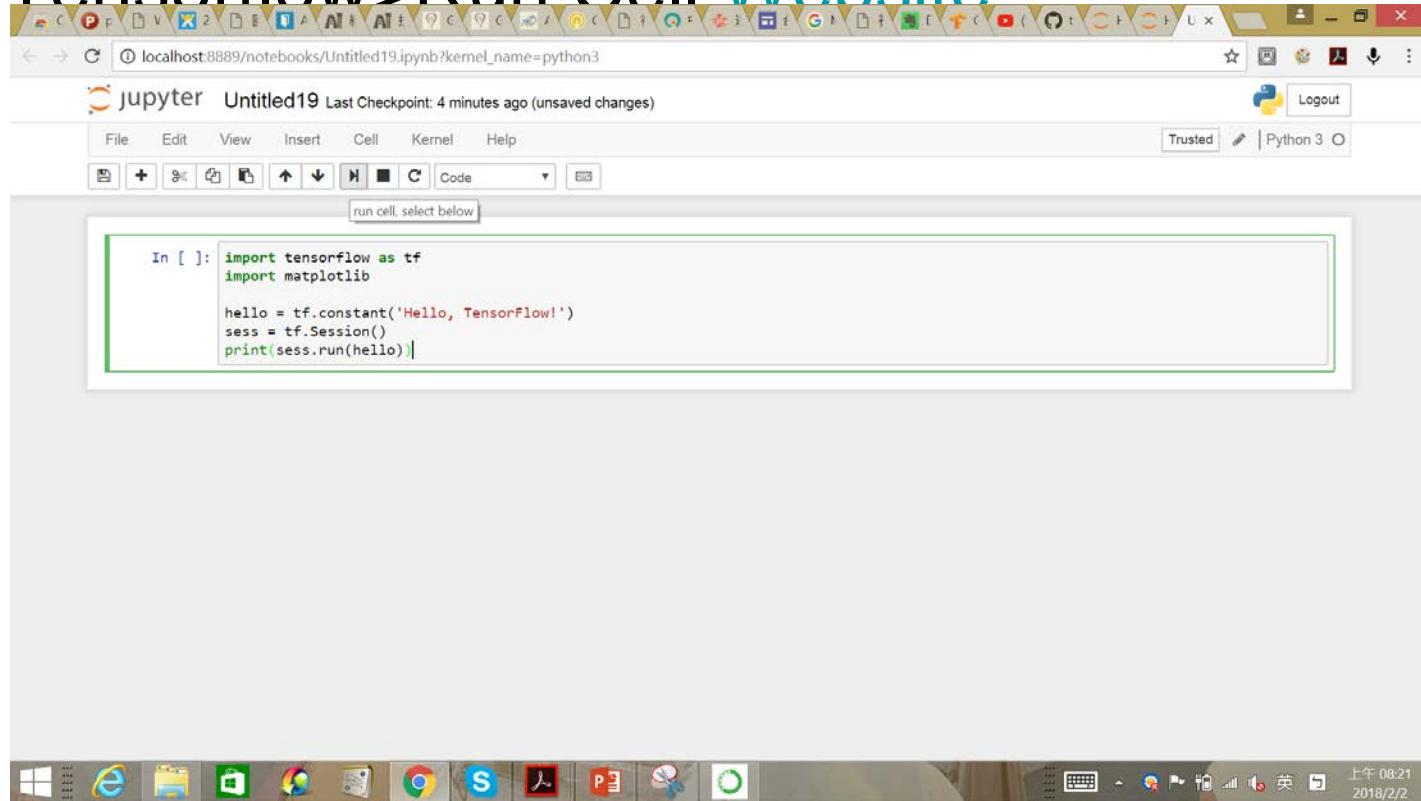
Jupyter->Launch



New->Python3



Test Tensorflow>Run Cell Website



Python Comments # (pound) sign

In [1]:

```
import tensorflow as tf
import matplotlib

hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

```
ModuleNotFoundError                         Traceback (most recent call last)
<ipython-input-1-ba59d47e36b0> in <module>()
      1 import tensorflow as tf
----> 2 import matplotlib
      3
      4 hello = tf.constant('Hello, TensorFlow!')
      5 sess = tf.Session()

ModuleNotFoundError: No module named 'matplotlib'
```

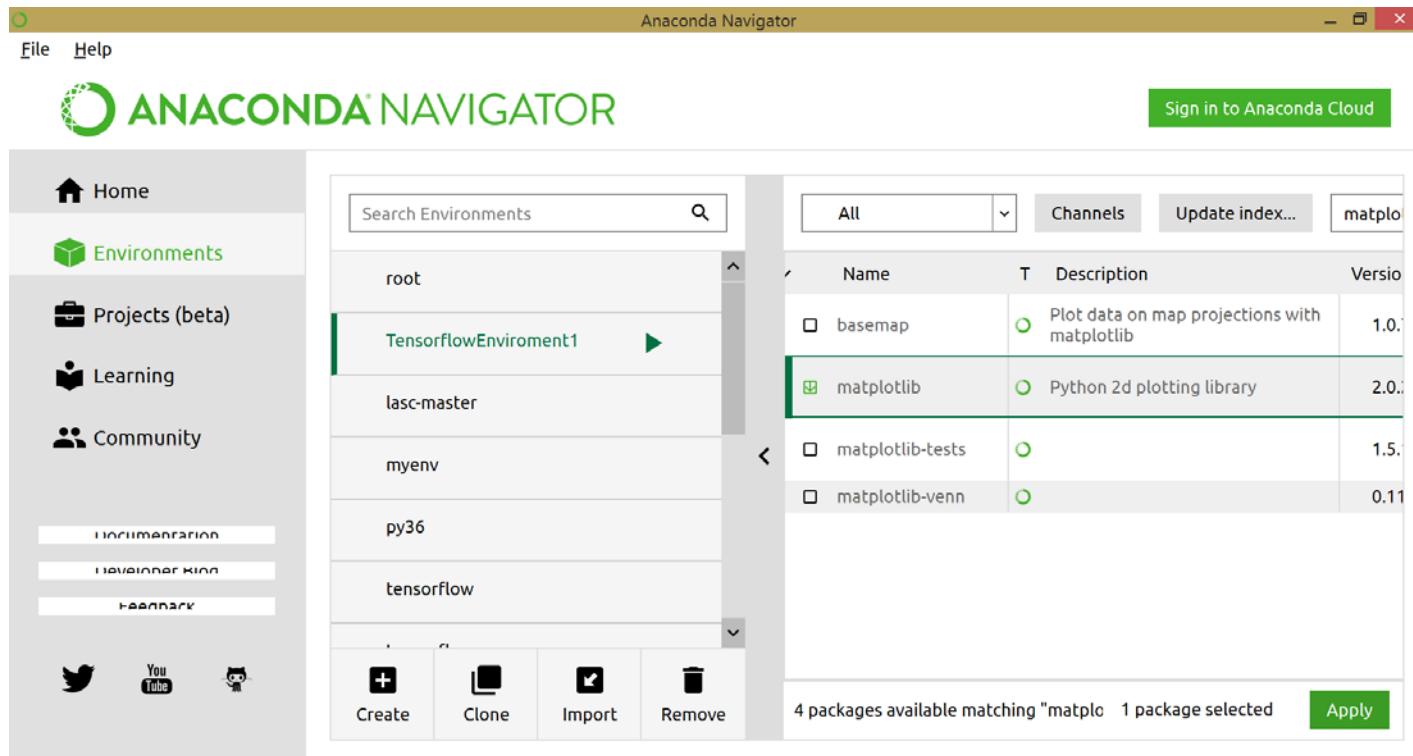
In [2]:

```
import tensorflow as tf
# import matplotlib

hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

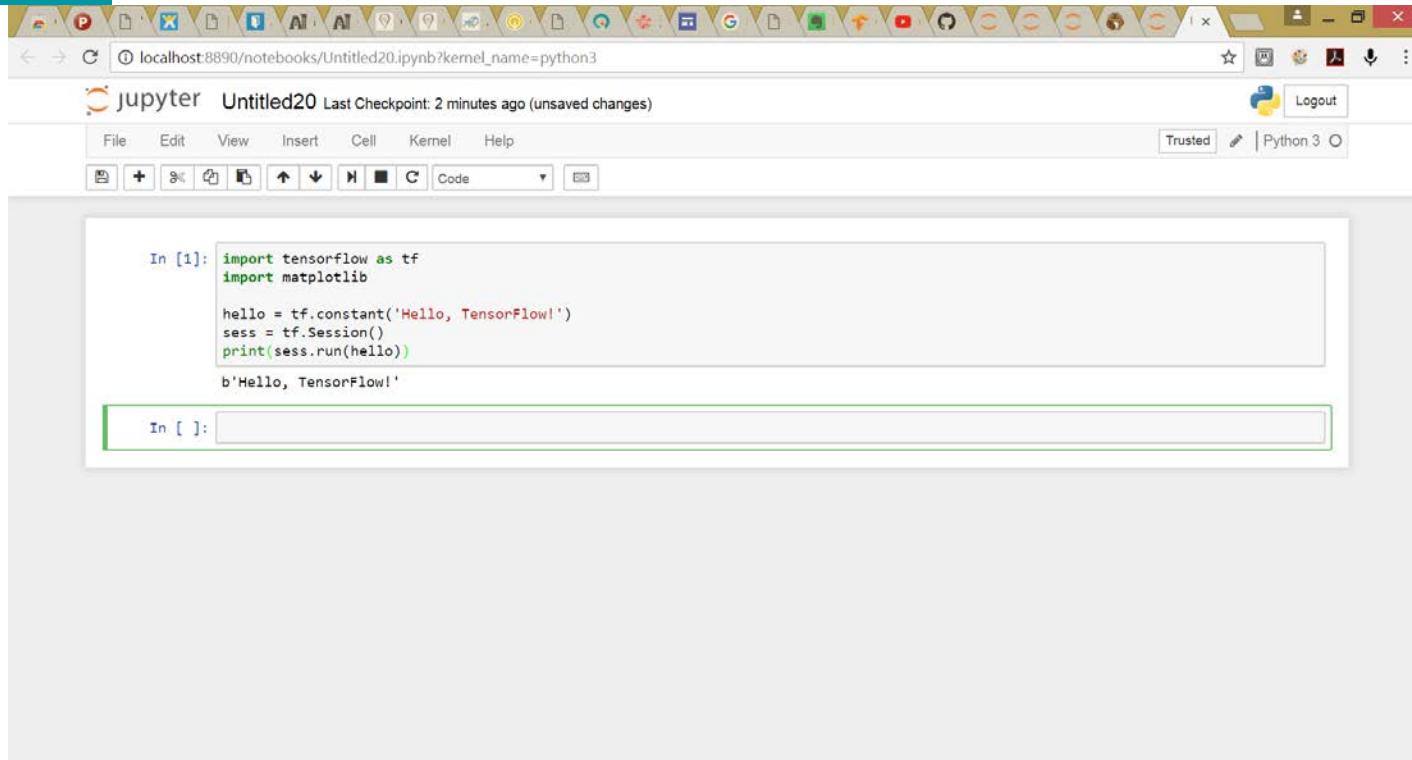
b'Hello, TensorFlow!'

Enviorment->matplotlib->apply



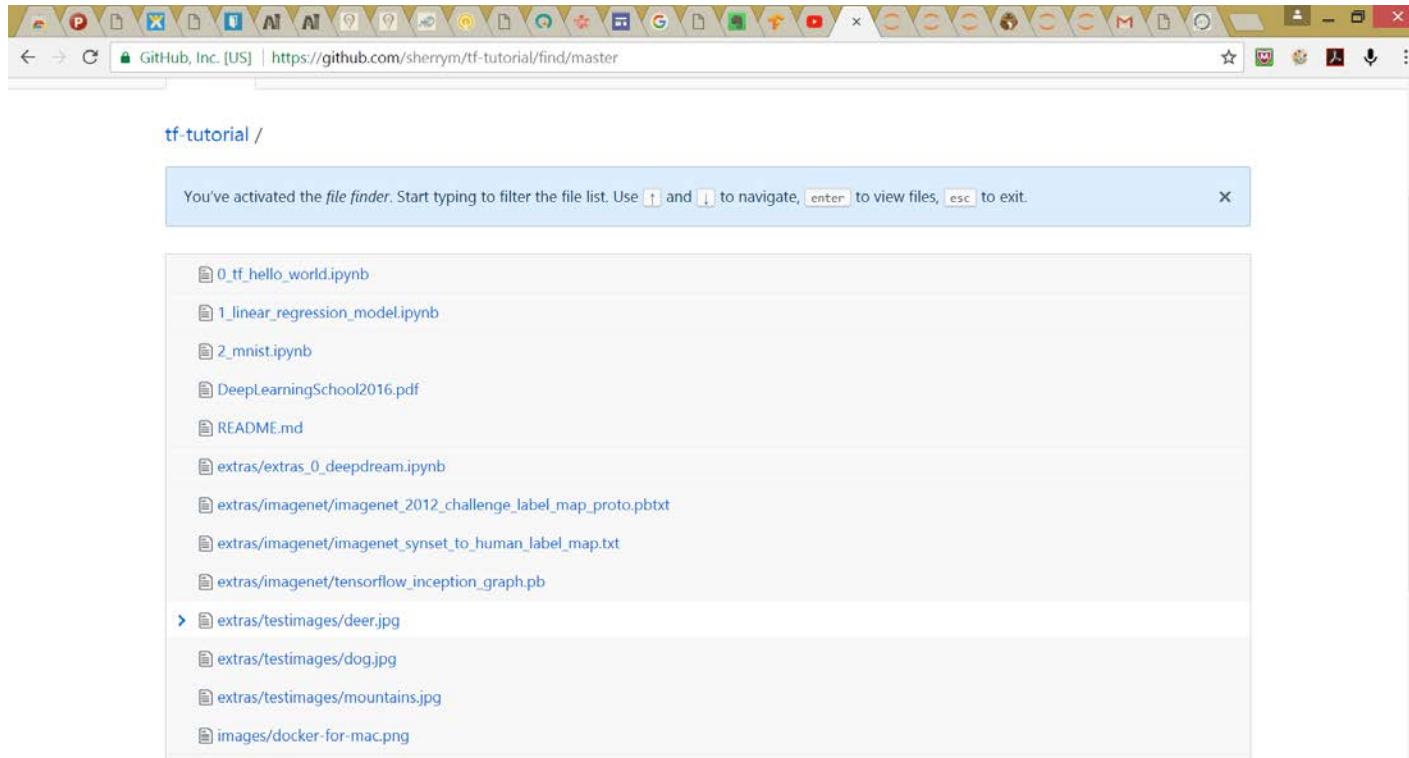
Home->Jupyter->Python3

Website

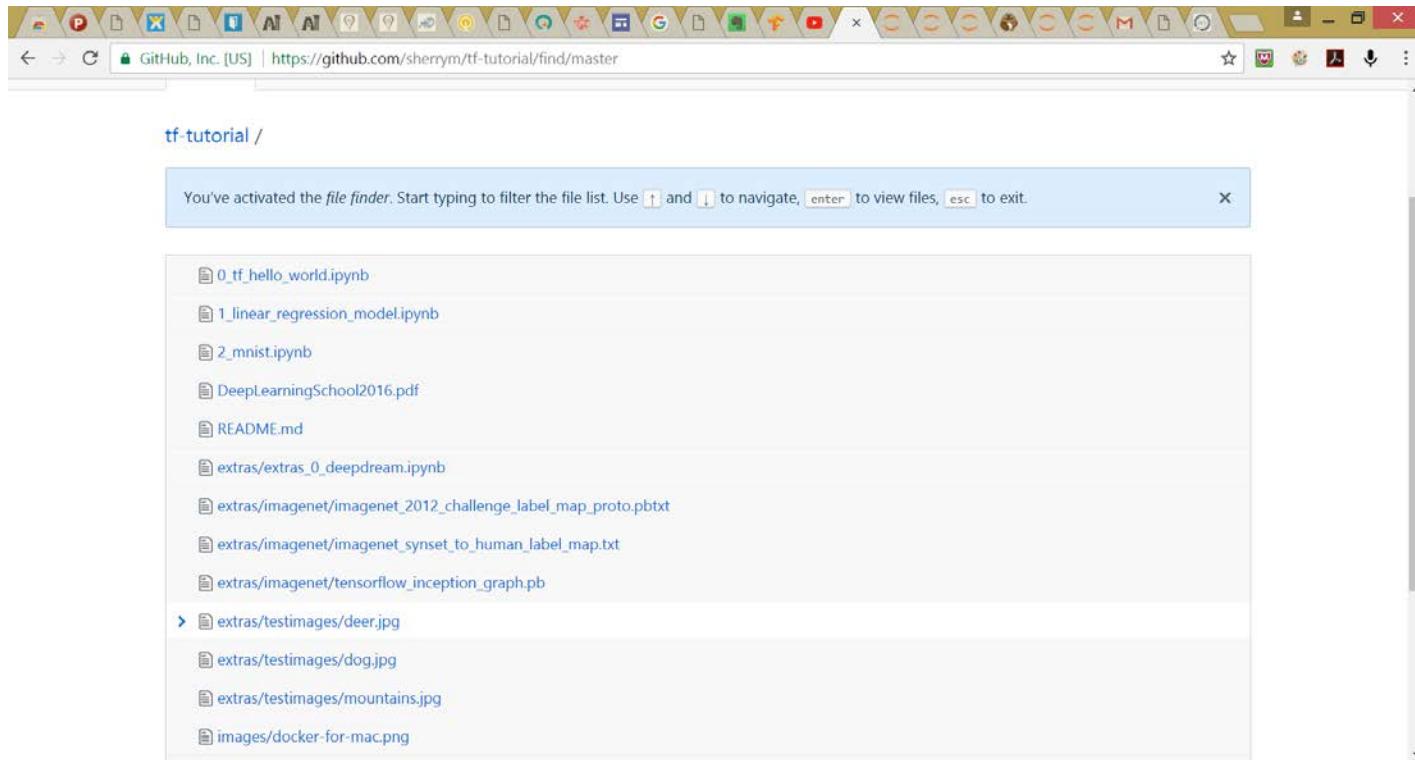


linear_regression_model.ipynb
[web site](#)

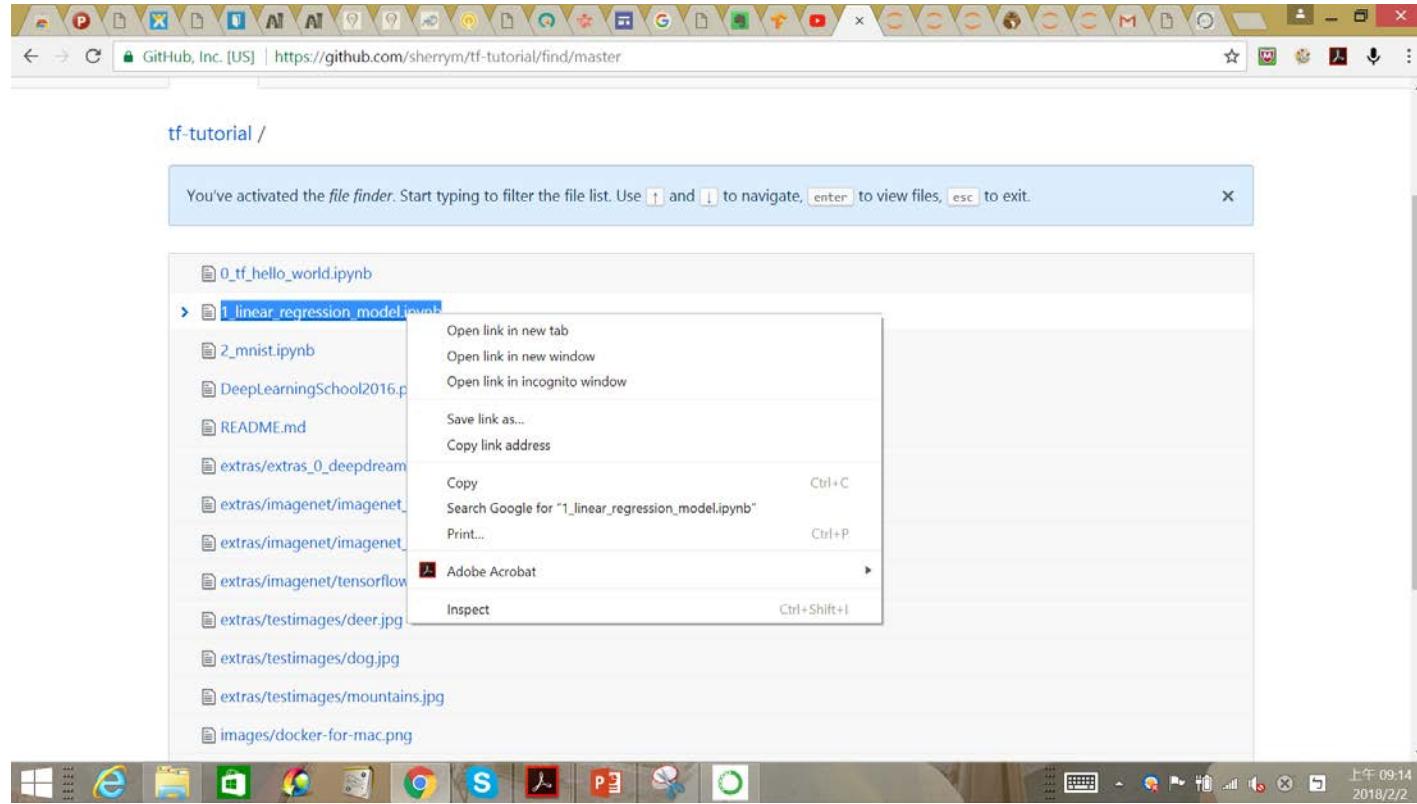
Find File



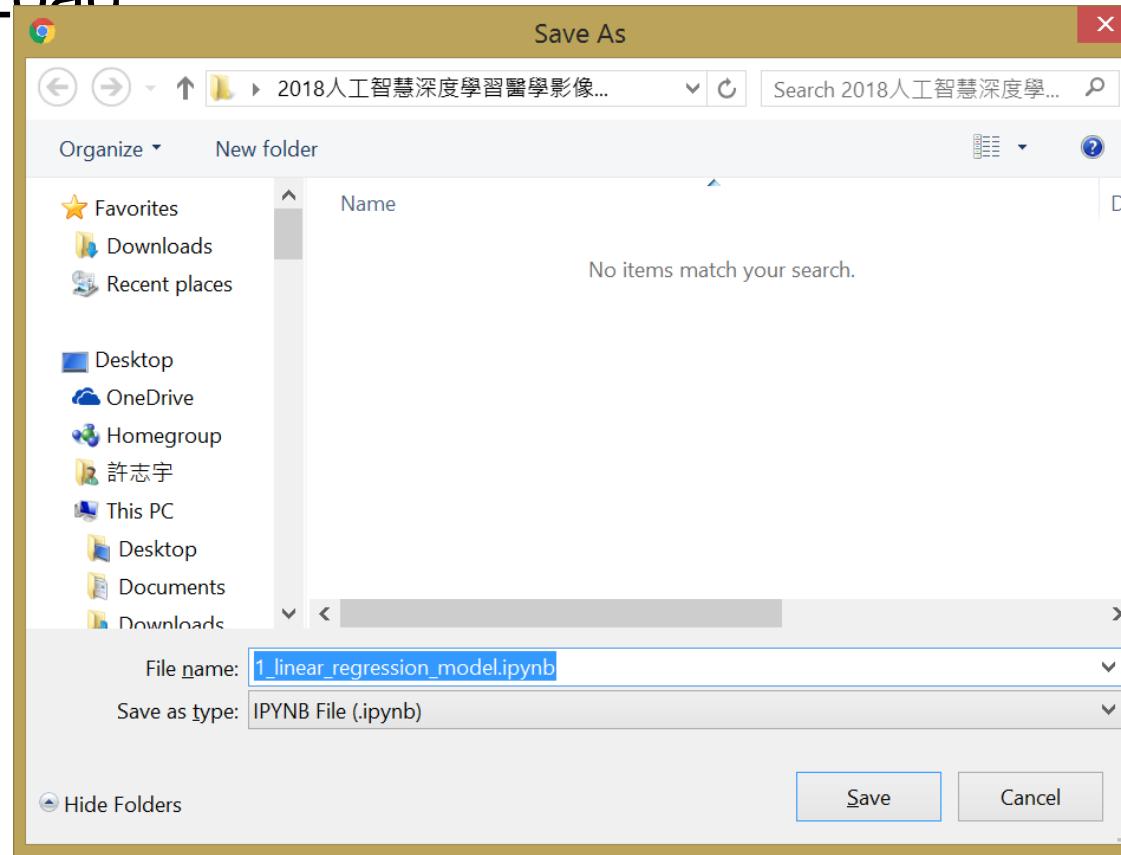
Download 1_linear_regression_model.ipynb



Right Mouse Button Click



Down Load



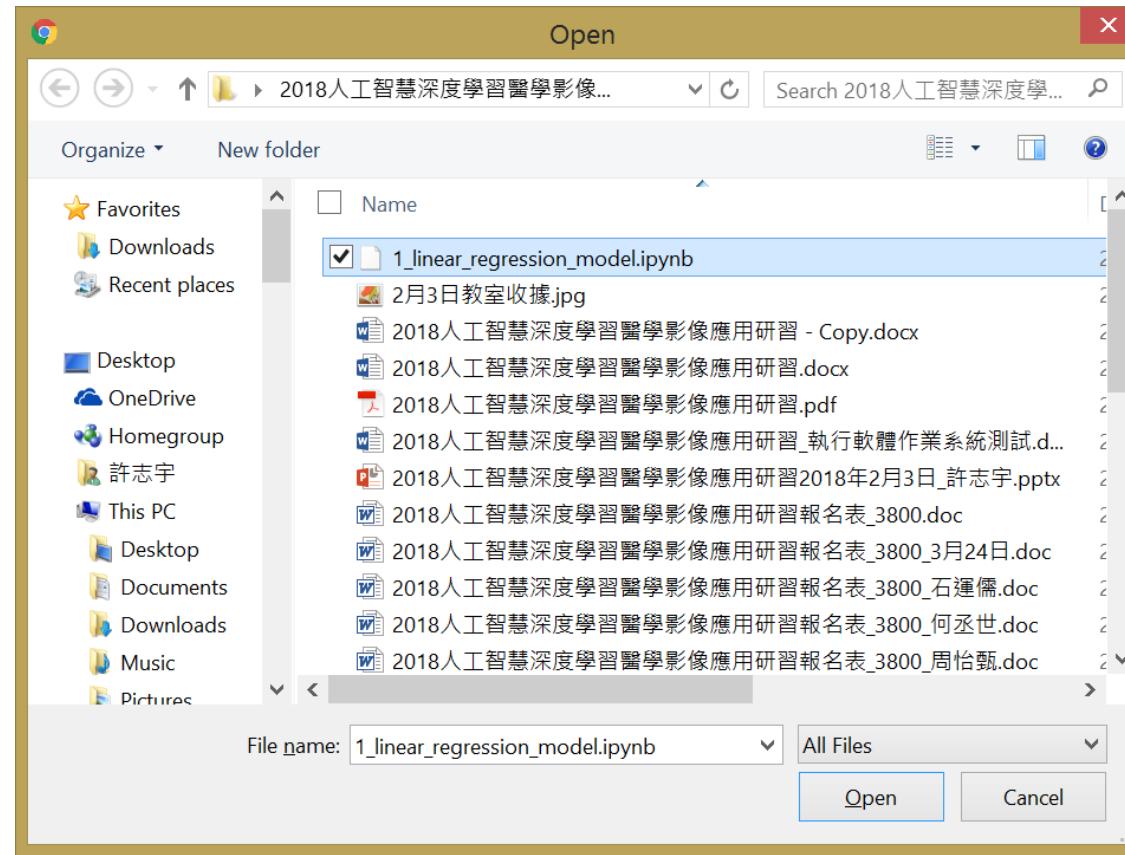
Upload

The screenshot shows a Jupyter Notebook interface running on a Windows operating system. The title bar indicates the URL is `localhost:8890/tree#`. The main area displays a file tree with the following contents:

Name	Last Modified
Anaconda3	7 days ago
AnacondaProjects	2 hours ago
Contacts	17 days ago
data	a month ago
Desktop	3 hours ago
Documents	3 days ago
Downloads	a minute ago
DSB2-master	2 years ago
Favorites	17 days ago
lasc-master	2 years ago
Links	13 days ago
MNIST_data	a month ago
Admin	17 days ago

At the top of the interface, there are tabs for "Files", "Running", and "Clusters". Below the tabs, a message says "Select items to perform actions on them." On the right side, there are buttons for "Upload", "New", and a dropdown menu. A text input field labeled "Name" with a browse button is also present. The bottom of the window shows the taskbar with various icons and the system tray indicating the date and time as "上午 09:16 2018/2/2".

Upload



Upload

The screenshot shows a Jupyter Notebook interface running in a web browser at `localhost:8890/tree#`. The title bar says "jupyter". The main area displays a file tree and an open notebook file.

File Tree:

- 1_linear_regression_model.ipynb (selected)
- Anaconda3
- AnacondaProjects
- Contacts
- data
- Desktop
- Documents
- Downloads
- DSB2-master
- Favorites
- lasc-master
- Links

Actions:

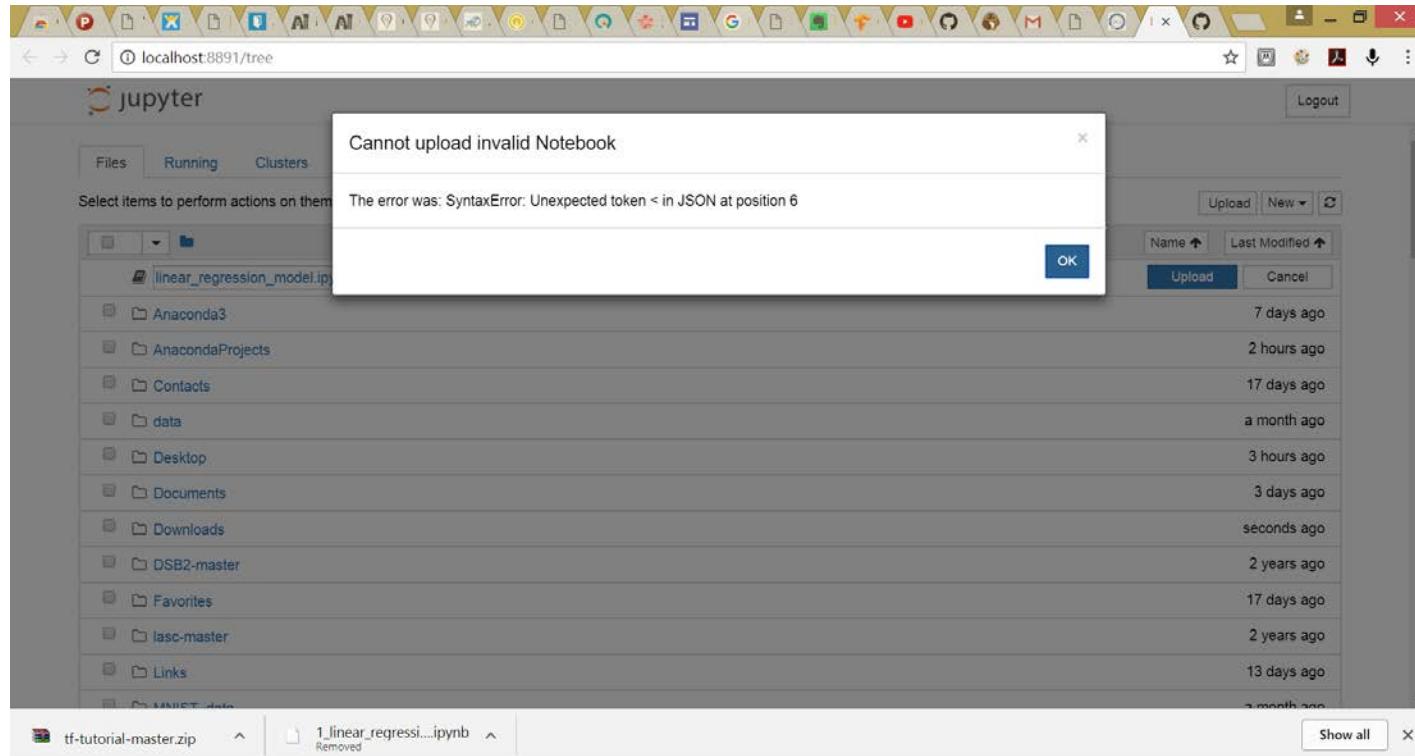
- Upload (button)
- New (button)
- Name ↑ (button)
- Last Modified ↑ (button)
- Cancel (button)

Open Notebook:

1_linear_regression_model.ipynb

Show all X

Upload



sherrym/tf-tutorial->clone

GitHub, Inc. [US] | <https://github.com/sherrym/tf-tutorial>

This repository Search Pull requests Issues Marketplace Explore

sherrym / tf-tutorial forked from wolfg/tf-tutorial

Watch 42 Star 315 Fork 342

Code Pull requests 0 Projects 0 Wiki Insights

Short tutorial for TensorFlow, designed to be presented in-person

41 commits 2 branches 0 releases 2 contributors

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

This branch is 1 commit behind wolfg:master.

sherrym Swapped slide order. Use GradientDescentOptimizer in example Latest commit 48b9971 on Sep 24, 2016

extras Add Pillow instructions a year ago

images Updated instructions a year ago

0_tf_hello_world.ipynb Added links back and forth a year ago

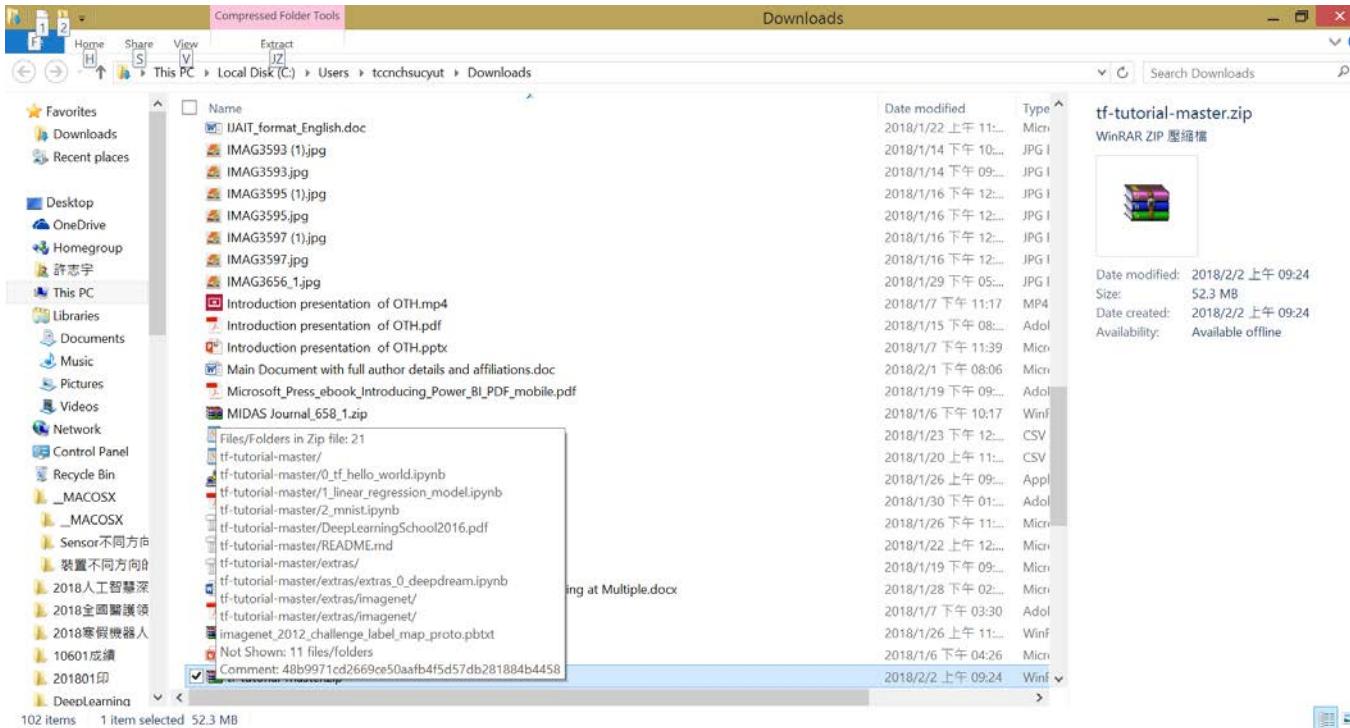
1_linear_regression_model.ipynb Swapped slide order. Use GradientDescentOptimizer in example a year ago

Fixed indentation a year ago

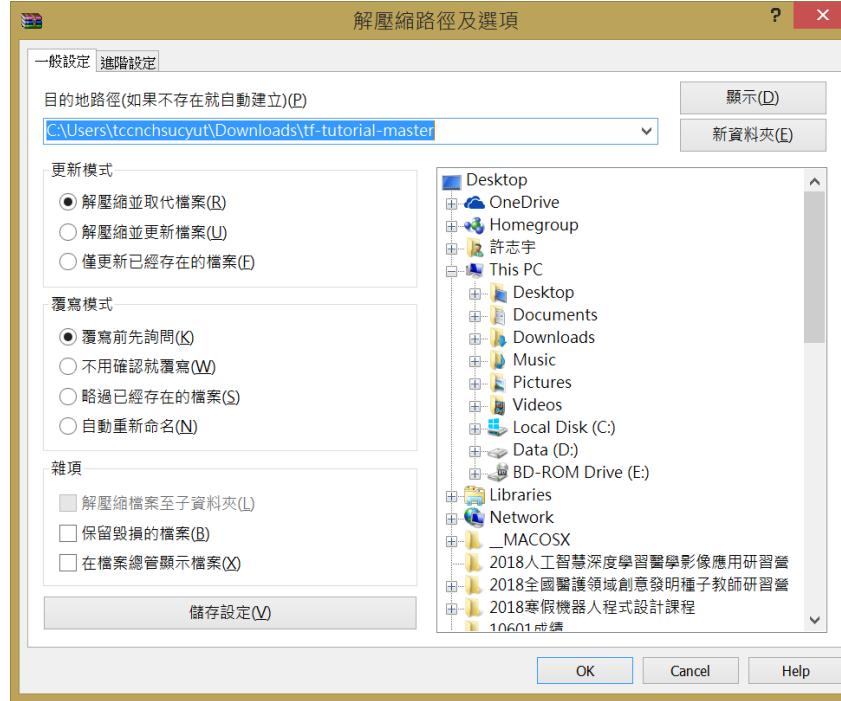
https://github.com/sherrym/tf-tutorial/releases

tf-tutorial-master.zip 1_linear_regression_model.ipynb Removed Show all

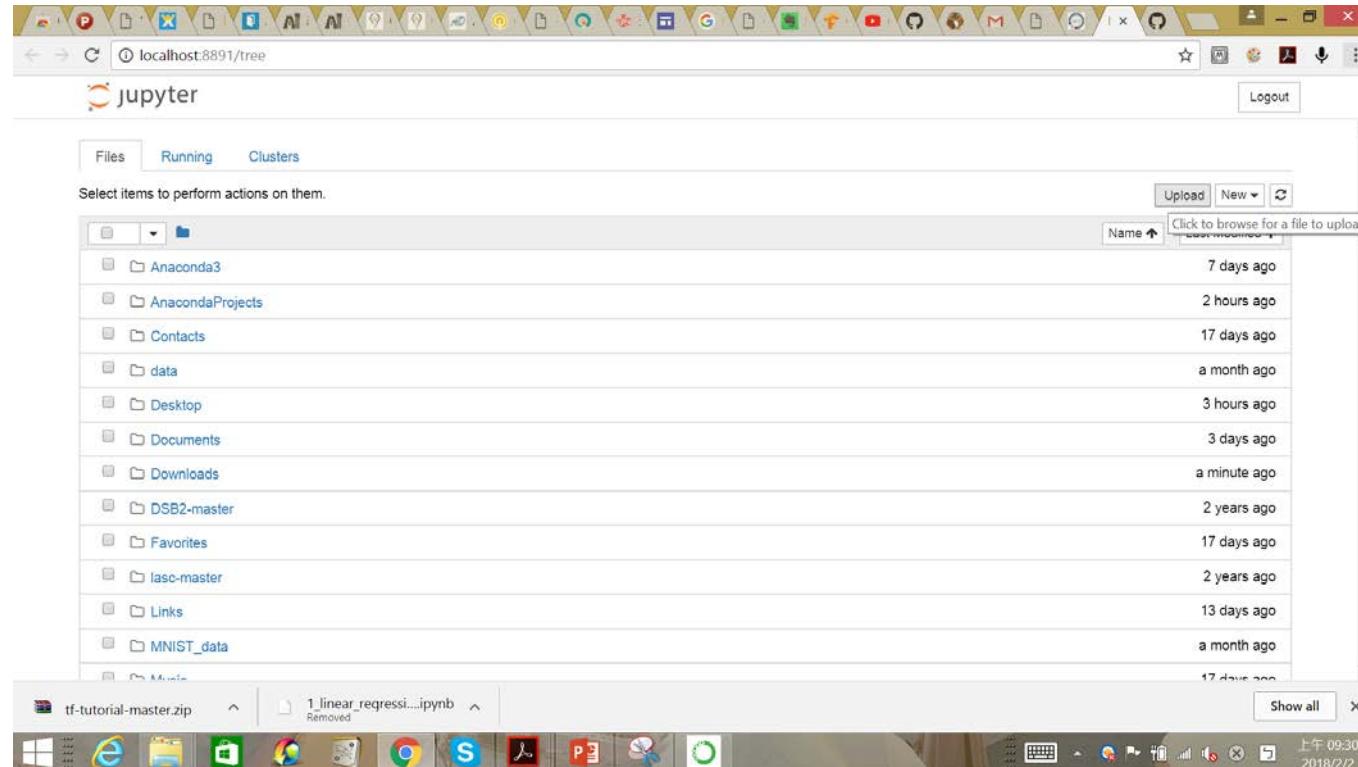
Extract file



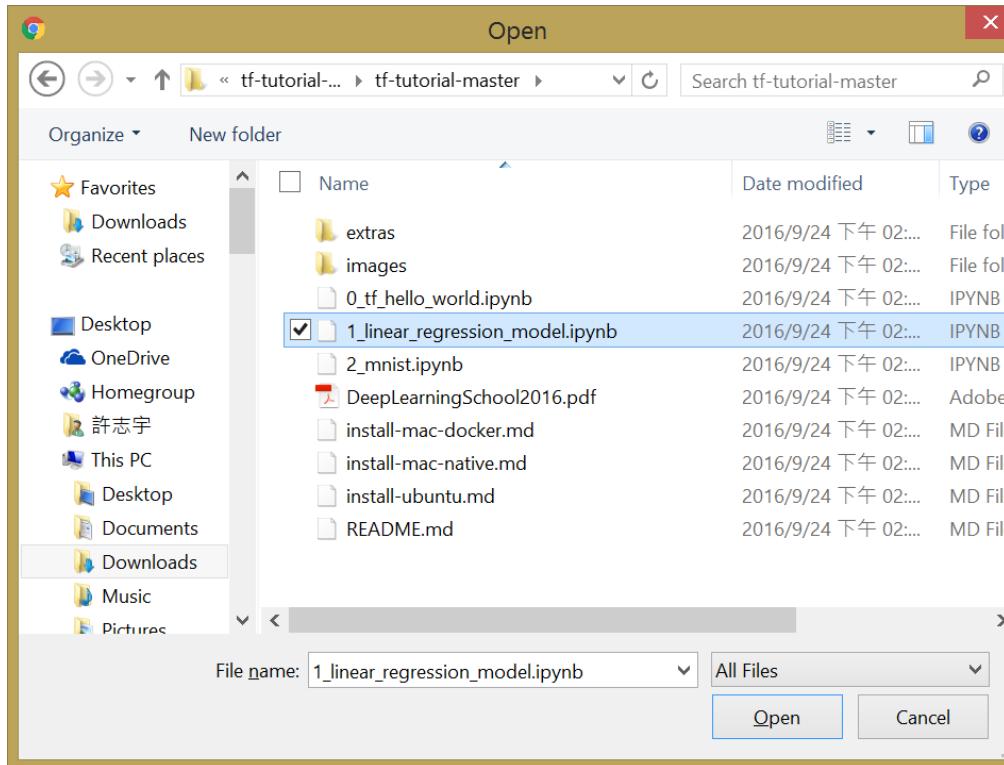
Extract file



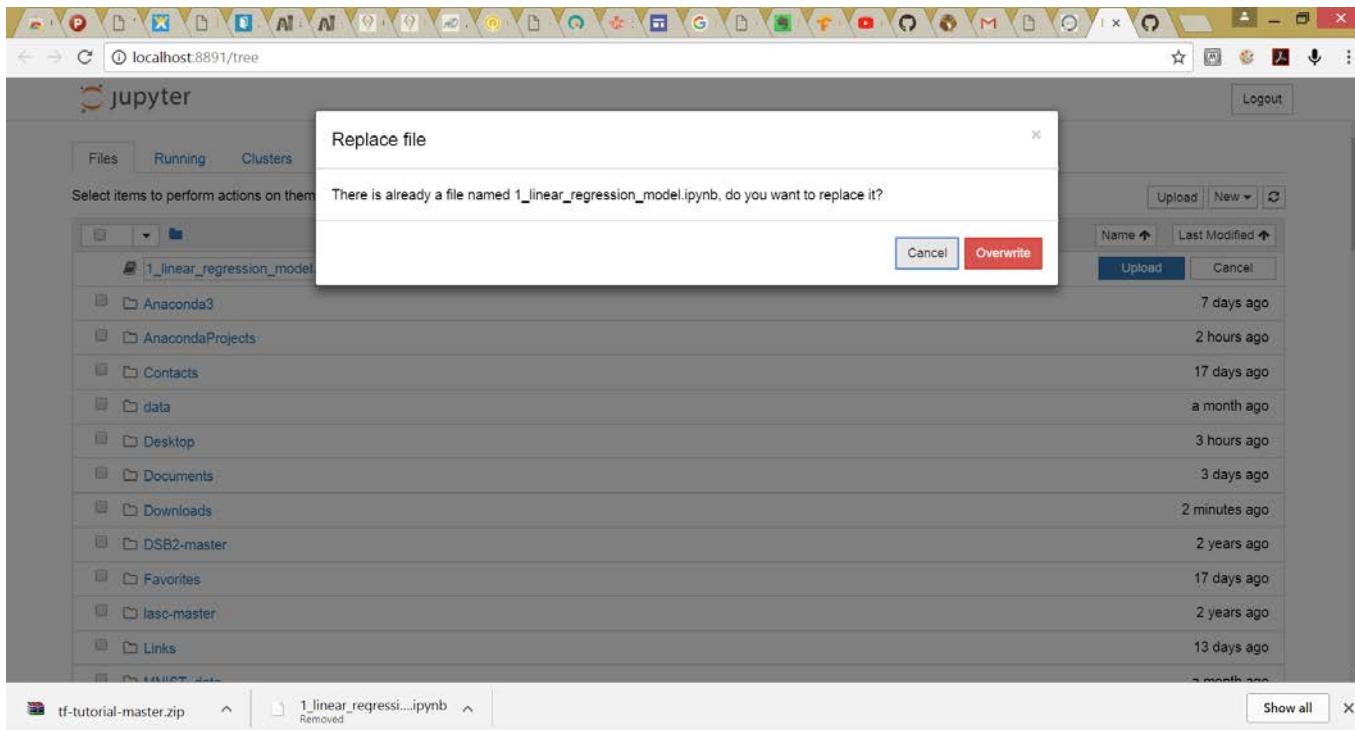
Upload



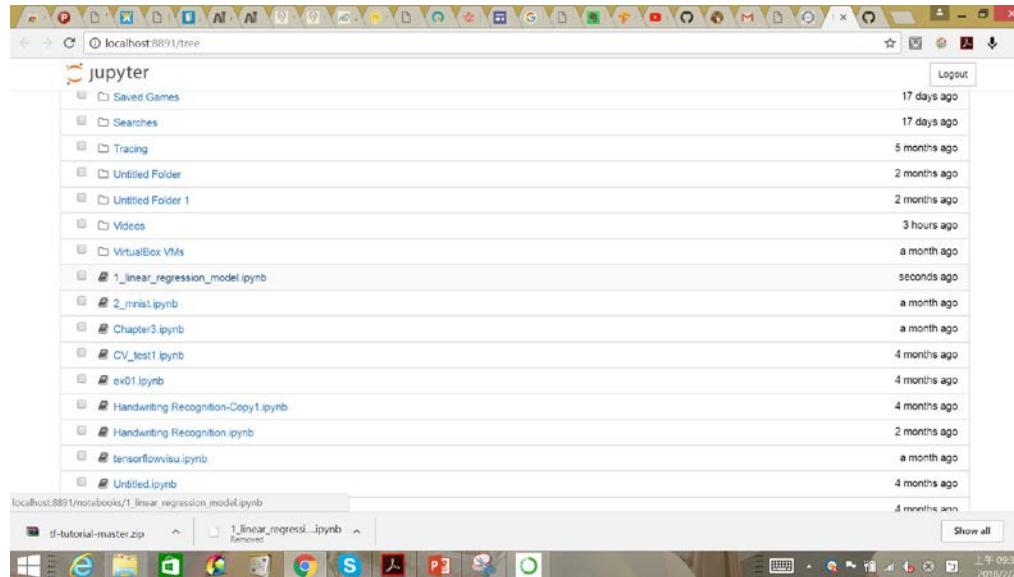
Upload



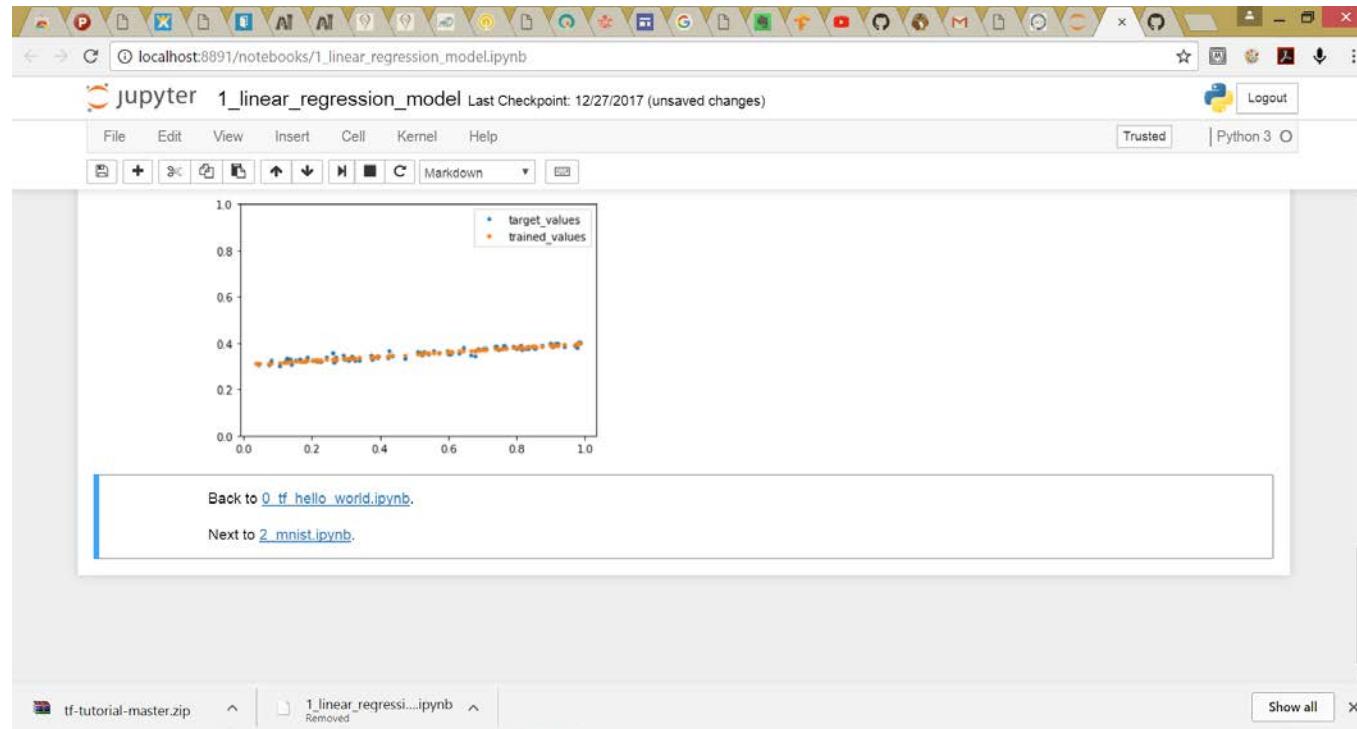
Overwrite



Open 1_linear_regression_model.ipynb



Run Cell



Regression Analysis

- How can we obtain a two dimensional line equation by given three points (x_1, y_1) , (x_2, y_2) and (x_3, y_3) ? A line can pass through only two points.

Regression Analysis

- There is no distance between a point and a line if the point is on the line.
- If three points are on the line $y=m*x+b$.
 $y_1=mx_1+b$
 $y_2=mx_2+b$
 $y_3=mx_3+b$
- On the other hand, there is some distance between a point and a line if the point is not on the line.
- How can we define the distance to easily obtain the optimal parameters m and b

Regression Analysis

- How can we define the object function to easily obtain the optimal parameters m and b without the definition of distance between a point and a line?
- This is a liner regression model problem.
- We do not use the definition of a distance between a point and a line.
- Instead of using distances between points and a line we use square errors defined by $(y_1-f(x_1))^2$, $(y_2-f(x_2))^2$ and $(y_3-f(x_3))^2$.

Regression Analysis

- The linear regression model problem is transformed to a optimal problem with a object function that is the sum of the square errors.
- The optimal parameters m and b are obtained as the object function is minimum.

Regression Analysis

- Partial derivatives of the error function respect to parameters m and b are named as a **gradient vector** and it is equal to **a zero vector** where critical points is located.
- We can obtain a vector equation.
- Gradient(sum of error function (m,b))=(0,0)The architecture of deep neural network(nn) is as follow,

What is a Model?

- A Model is a mathematic function with parameters and inputs and outputs.

What are the parameters for linear equations?

- For example, a linear equation.
- $y=m*x+b$ of a line in two dimensional coordinate
- the slope m and intercept b are parameters.

A model is a function

- If we define a function
 $f(x)=mx+b$ and $y=f(x)$
x is input and y=output.

Parameters

- We can obtain any line by setting the parameters m and b .

What is learning?

- Unsupervised Learning
 - Machine can automatically choose the parameters by data set.
- Supervised Learning
 - Machine can automatically choose the parameters by data set and Labels.

What is learning?

- For example, how could we decide a two dimensional line equation by given two points(x_1, y_1) and (x_2, y_2)?
- Two points give one data set

What is learning?

- We substitute two pair coordinates into the equation
- $y=mx+b$ to solve the simultaneous linear equations to obtain parameters m and b.

$$y_1 = m \cdot x_1 + b$$

$$y_2 = m \cdot x_2 + b$$

- These two equations can be used to derive the parameters m and b.

$$m = (y_2 - y_1) / (x_2 - x_1)$$

$$b = y_1 - (y_2 - y_1) / (x_2 - x_1) \cdot x_1$$

- We can compute m and b by above two formulas.
- Analytic Solution or Close form of parameters

What is learning?

- Could we obtain parameters m and b automatically by machine?
- Define Function $f(x)=mx+b$

$$f(x_1)=m \cdot x_1 + b$$

$$f(x_2)=m \cdot x_2 + b$$

- The data set are (x_1, y_1) and (x_2, y_2)
- The square error between $f(x_1)$ and y_1 is defined as $(f(x_1) - y_1)^{**2}$
- The square error between $f(x_2)$ and y_2 is defined as $(f(x_2) - y_2)^{**2}$
- The sum of square function is defined as
- $\text{SumErrors}(m, b) = (m \cdot x_1 + b - y_1)^{**2} + (m \cdot x_2 + b - y_2)^{**2}$

What is learning?

- Take partial deferential operation on $\text{SumErrors}(m,b)$ respect to m. (Chain Rule)
- $2(m*x_1+b-y_1)*x_1 + 2(m*x_2+b-y_2)*x_2$
- Take partial deferential operation on $\text{SumErrors}(m,b)$ respect to b.) (Chain Rule)
- $2(m*x_1+b-y_1) + 2(m*x_2+b-y_2)$
- For finding optimum, we have to set them to zero .
- $2(m*x_1+b-y_1)*x_1 + 2(m*x_2+b-y_2)*x_2=0$
- $2(m*x_1+b-y_1) + 2(m*x_2+b-y_2)=0$

What is learning?

- $2(m*x_1+b-y_1)*x_1 + 2(m*x_2+b-y_2)*x_2 = 0$
- $2(m*x_1+b-y_1) + 2(m*x_2+b-y_2) = 0$
- Transform
- $2(m*x_2+b-y_2)*(x_2-x_1) = 0$
- $2(m*x_1+b-y_1)*(x_1-x_2) = 0$
- Transform
- $mx_2+b-y_2=0 \Rightarrow y_2=mx_2+b$
- $mx_1+b-y_1=0 \Rightarrow y_1=mx_1+b$
- These two equations can be used to derive the parameters m and b.

$$m = (y_2 - y_1) / (x_2 - x_1)$$

$$b = y_1 - (y_2 - y_1) / (x_2 - x_1) * x_1$$

- We can compute m and b by above two formulas.
- Analytic Solution or Close form of parameters

What is learning?

- How can we obtain a two dimensional line equation by given three points (x_1, y_1) , (x_2, y_2) and (x_3, y_3) ?
- Could we obtain analytic solutions of parameters m and b in Close form.
- Could we can compute m and b by two formulas?
- Yes or No, home work1!
- Home work2, repeat home work1, Please replace error function by distance function