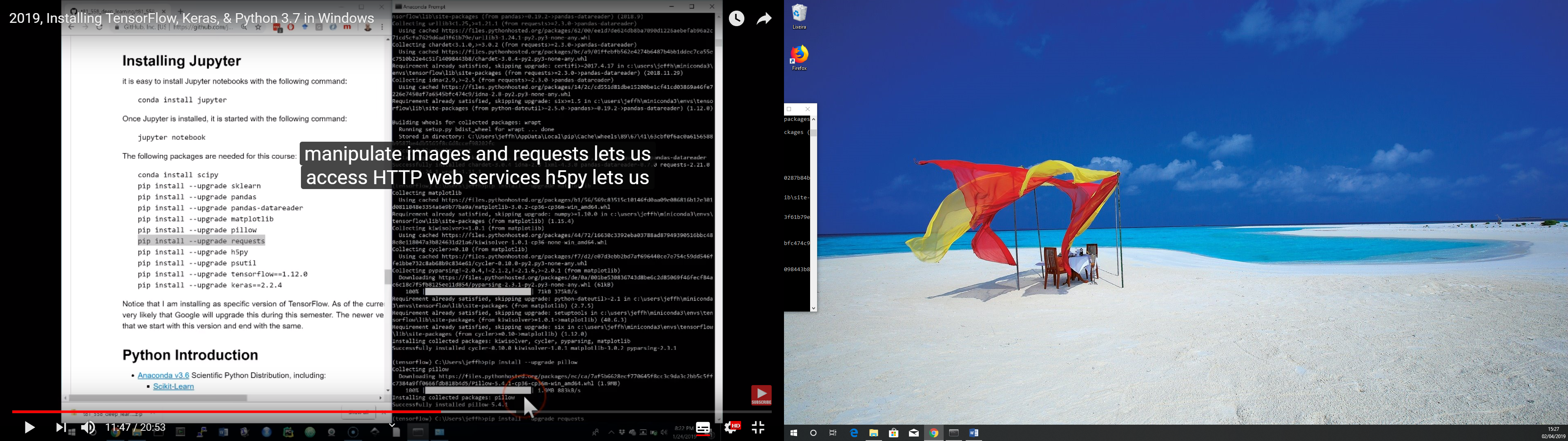
Tuturial:

<https://www.youtube.com/watch?v=59duINoc8GM>

Conda create –name tensorflow python=3.6



Python

import tensorflow as tf

print(tf.\_\_version\_\_)

python -m ipykernel install --user --name tensorflow --display-name "Python 3.6 (tensorflow)"

activate tensorflow

import numpy as np

(train\_data, train\_labels), (test\_data, test\_labels) = reuters.load\_data(num\_words=10000)

print(len(train\_data), 'train sequences')

print(len(test\_data), 'test sequences')

print(train\_data[10])

word\_index=reuters.get\_word\_index()

reverse\_word\_index=dict([(value,key) for(key,value) in word\_index.items()])

decoder\_newswire=' '.join([reverse\_word\_index.get(i-3,'?') for i in train\_data[0]])

print(decoder\_newswire)

num\_classes = np.max(train\_labels)+1

print(num\_classes, 'classes')

from keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer(num\_words=10000)

train\_data=tokenizer.sequences\_to\_matrix(train\_data,mode='binary')

test\_data=tokenizer.sequences\_to\_matrix(test\_data,mode='binary')

from keras.utils.np\_utils import to\_categorical

one\_hot\_train\_labels = to\_categorical(train\_labels, num\_classes)

one\_hot\_test\_labels = to\_categorical(test\_labels, num\_classes)

>>> from keras import models

>>> from keras import layers

>>> model = models.Sequential()

>>> model.add(layers.Dense(64, activation='relu', input\_shape=(10000,)))

model.add(layers.Dense(64, activation='relu'))

model.add(layers.Dense(46, activation='softmax'))

model.compile(optimizer='rmsprop', loss='categorical\_crossentropy', metrics=['accuracy'])

history = model.fit(train\_data, train\_labels, batch\_size=512, epochs=20, verbose=1, validation\_split=0.1)