Code: TSNE

import numpy as np

import matplotlib.pyplot as plt

from itertools import product

from sklearn.manifold import TSNE

from sklearn.utils import shuffle

from sklearn.datasets import fetch\_mldata

from sklearn.manifold import Isomap

from sklearn.manifold import MDS

final\_mnist = fetch\_mldata("MNIST Original")

ourtargets = list(range(0,10))

XX\_train, yy\_train = final\_mnist.data / 255., final\_mnist.target

X\_train=[]

y\_train=[]

for i, label in enumerate(yy\_train):

if label in ourtargets:

X\_train.append(XX\_train[i])

y\_train.append(yy\_train[i])

num\_samples\_to\_plot = 5000

X\_train, y\_train = shuffle(X\_train, y\_train)

X\_train, y\_train = X\_train[:num\_samples\_to\_plot], y\_train[:num\_samples\_to\_plot]

for digit in ourtargets:

instances=[i for i in y\_train if i==digit]

print ("Digit", digit,"appears ",len(instances), "times")

transformer = TSNE(n\_components = 2, perplexity=40, verbose=2)

X\_transformed = transformer.fit\_transform(X\_train)

plot.scatter(X\_transformed[:, 0], X\_transformed[:, 1], c=y\_train)

#ISOMAP

data\_y = final\_mnist.data[::30]

target = final\_mnist.target[::30]

K = Isomap(n\_components=2, n\_neighbors=10)

proj = K.fit\_transform(data\_y)

plt.scatter(proj[:, 0], proj[:, 1], c=target, cmap=plt.cm.get\_cmap('jet', 10))

plt.colorbar(ticks=range(10))

plt.clim(-0.5, 9.5)

plt.show()

count=0;

for label , x, y in zip(y\_train, X\_transformed[:, 0], X\_transformed[:, 1]):

#Lets annotate every 1 out of 200 samples, otherwise graph will be cluttered with anotations

if count % 200 == 0:

plt.annotate(str(int(label)),xy=(x,y), color='black', weight='normal',size=10,bbox=dict(boxstyle="round4,pad=.5", fc="0.8"))

count = count + 1

plt.tight\_layout()

plt.suptitle("TSNE for MNIST digits ")

plt.show()