$_path ='$

$/home/hpc//mnist_tutorial/mnist'$

$_datamnist =$

$input_data.read_data_sets(data_path, one_hot =$

$False)$

$_rate =$

$0.01training_epochs =$

$5batch_size =$

$256display_step =$

$1examples_to_show =$

10

$_input =$

$784MNISTdatainput(imgshape :$

$28*$

$28)$

$_input])$

$_hidden_1 =$

$256n_hidden_2 =$

$128weights =$

$'encode_h1' : tf.Variable(tf.random_normal([n_input, n_hidden_1])),' encode_h2' : tf.Variable(tf.random_normal([n_hidden_1, n$

$'encode_h1' : tf.Variable(tf.random_normal([n_hidden_1])),' encode_h2' : tf.Variable(tf.random_normal([n_hidden_2])),' decod$

$_1 =$

$tf.nn.sigmoid(tf.add(tf.matmul(x, weights['encode_h1']), bias['encode_h1']))layer_2 =$

$tf.nn.sigmoid(tf.add(tf.matmul(layer_1, weights['encode_h2']), bias['encode_h2']))returnlayer_2$

$_1 =$

$tf.nn.sigmoid(tf.add(tf.matmul(x, weights['decode_h2']), bias['decode_h2']))layer_2 =$

$tf.nn.sigmoid(tf.add(tf.matmul(layer_1, weights['decode_h1']), bias['decode_h1']))returnlayer_2$

$_op =$

$encode(x)decode_op =$

$decode(encode_op)y_pred =$

$decode_opy_true =$

$xcost =$

$tf.reduce_mean(tf.square(y_pred-$

$y_true))optimizer =$

$tf.train.AdamOptimizer(learning_rate).minimize(cost)$

$_variables_initializer()sess.run(init)total_batch =$

$int(mnist.train.num_examples/batch_size)forepochinrange(training_epochs) :$

$foriinrange(total_batch) :$

$batch_xs, batch_ys =$

$mnist.train.next_batch(batch_size)_,c =$

$sess.run([optimizer, cost], feed_dict =$

$x : batch_xs)ifepochprint("Epoch :$

$",' print('Optimizefinish')encode_decode =$

$sess.run(y_pred, feed_dict =$

$x : mnist.test.images[: examples_to_show])f, a =$

$plt.subplots(2, 10, figsize =$

$(10, 2))foriinrange(examples_to_show) :$

$a[0][i].imshow(sess.run(tf.reshape(mnist.test.images[i], [28, 28])))a[1][i].imshow(sess.run(tf.reshape(encode_decode[i],$

$800)$

$_encode.png$

$_datapath ='$

$/home/hpc//mnist_tutorial/mnist'mnist =$

$input_data.read_data_sets(path, one_hot =$

$False)$

$_rate =$

$0.01training_epochs =$

$5batch_size =$

$256display_step =$

$1examples_to_show =$

10

$_input =$

$784MNISTdatainput(imgshape :$

$28*$

$28)$

$_input])$

$_hidden_1 =$

$2561stlayernumfeaturesn_hidden_2 =$

$1282ndlayernumfeatures$

$_rate =$

$0.010.01thislearningratewillbebetter!Testedtraining_epochs =$

$10batch_size =$

$256display_step =$

$1n_input =$

$784MNISTdatainput(imgshape :$

$28*$

$28)X =$

$tf.placeholder("float", [None, n_input])n_hidden_1 =$

$128n_hidden_2 =$

$64n_hidden_3 =$

$10n_hidden_4 =$

$2weights =$

$'encoder_h1' : tf.Variable(tf.truncated_normal([n_input, n_hidden_1],)),' encoder_h2' : tf.Variable(tf.truncated_normal([n_hi$

$'encoder_b1' : tf.Variable(tf.random_normal([n_hidden_1])),' encoder_b2' : tf.Variable(tf.random_normal([n_hidden_2])),' enc$