

Answer to Exercise 1 (the line numbers are for reference only. They may be slightly different in your case)

Make the following changes to “mnist_tutorial_tf.py”:

Line 18 (insert): `import os`

Line 23 (replace): `from cleverhans.utils_tf import model_eval, tf_model_load`

Line 25 (replace): `from cleverhans.attacks import FastGradientMethod, CarliniWagnerL2`

Lines 37 (insert, “MODE”): `MODEL_PATH = 'models'` (see tutorial of Week 11):

`MODEL_PATH = 'models'`

`MODEL_ADV_PATH = 'models'`

Lines 41 (replace):

```
def mnist_tutorial(train_session, test_session, test_start=0,
                    test_end=100, HS, batch_size=BATCH_SIZE,
                    learning_rate=LEARNING_RATE,
                    clean_train=CLEAN_TRAIN,
                    testing=False,
                    backprop_through_attack=BACKPROP_THROUGH_ATTACK,
                    nb_filters=NB_FILTERS, num_threads=None,
                    model_path=MODEL_PATH,
                    model_adv_path=MODEL_ADV_PATH,
                    label_smoothing=0.1):
```

Lines 136 (replace):

```
if os.path.exists(model_path + ".meta"):
    tf_model_load(sess, model_path)
else:
    train(sess, loss, x_train, y_train, evaluate=evaluate,
          args=train_params, rng=rng, var_list=model.get_params())
    saver = tf.train.Saver()
    saver.save(sess, model_path)
```

Lines 226 (insert):

```
flags.DEFINE_string('model_path', MODEL_PATH,
                    'Path to save or load the model trained on clean examples')
flags.DEFINE_string('model_adv_path', MODEL_ADV_PATH,
                    'Path to save or load the model trained on adversarial samples')
```

Answer to Exercise 2 (the line numbers are for reference only. They may be slightly different in your case)

1. In order to get the first image, make the following changes to “mnist_tutorial_cw.py”:
 - (1) Line 38 (replace): `TARGETED = False`
2. In order to get the second image, make the following changes to “mnist_tutorial_cw.py”:
 - (1) Line 18 (replace): `from cleverhans.attacks import CarliniWagnerL2, FastGradientMethod`
 - (2) Line 192 (insert):

```
fgsm_params = {
    'eps': 0.3,
    'clip_min': 0.,
    'clip_max': 1.
}
fgsm = FastGradientMethod(model, sess=sess)
adv_x = fgsm.generate(x, **fgsm_params)
adv_image = adv_x.eval(session=sess, feed_dict={x: adv_inputs})
```

- (3) Line 221: `grid_viz_data[j, 1] = adv_image[j]`