1. Lines 191-216 (the line numbers are for reference only. They may be slightly different in your case):

```
if os.path.exists(model_adv_path + ".meta"):
  tf_model_load(sess, model_adv_path)
else:
  train(sess, loss2, x_train, y_train, evaluate=evaluate2,
      args=train_params, rng=rng, var_list=model2.get_params())
  saver = tf.train.Saver()
  saver.save(sess, model_adv_path)

evaluate2()
source_samples = test_end
attack_iterations = 50
CW_LEARNING_RATE = .2
adv_inputs = x_test[:source_samples]
adv_ys = None
yname = "y"
cw_params = {'binary_search_steps': 1,
         yname: adv_ys,
         'max_iterations': attack_iterations,
         'learning_rate': CW_LEARNING_RATE,
         'batch_size': source_samples,
         'initial_const': 10}
cw = CarliniWagnerL2(model2, sess=sess)
adv = cw.generate_np(adv_inputs, **cw_params)
eval_params = {'batch_size': np.minimum(nb_classes, source_samples)}
adv_accuracy = model_eval(sess, x, y, preds2, adv, y_test[:source_samples], args=eval_params)
print('Test accuracy on adv. examples generated by C&W: {0:.4f}'.format(adv_accuracy))
```