

# 深度学习方法与实践第二次作业

姓名：杨玉雷

学号：18023040

## 1. 基础作业

假设有函数  $y = \cos(ax + b)$ ，其中  $a$  为学号前两位， $b$  为学号最后两位。首先从此函数中以相同步长（点与点之间在  $x$  轴上距离相同），在  $0 < (ax+b) < 2\pi$  范围内，采样出 2000 个点，然后利用采样的 2000 个点作为特征点进行三次函数拟合（三次函数形式为  $y = w_1 * x + w_2 * x^2 + w_3 * x^3 + b$ ，其中  $w_i$  为可训练的权值， $b$  为可训练的偏置值， $x$  和  $y$  为输入的训练数据）。要求使用 TensorFlow 实现三次函数拟合的全部流程。拟合完成后，分别使用 ckpt 模式和 PB 模式保存拟合的模型。然后，针对两种模型存储方式分别编写恢复模型的程序。两个模型恢复程序分别使用 ckpt 模式和 PB 模式恢复模型，并将恢复的模型参数（ $w_i$  和  $b$ ）打印在屏幕上，同时绘制图像（图像包括所有的 2000 个采样点及拟合的函数曲线）。

请提交文档，内容包括模型参数配置、程序运行截图、拟合的三次函数以及绘制的图像，同时提交三个 python 脚本文件：1. 函数拟合及两种模式的模型保存程序 2. ckpt 模型恢复程序，3. PB 模型恢复程序。

## 实验过程和结果

### 1.1 函数拟合三次曲线

本实验按如下流程进行实验，并附上关键代码。

#### (1) 定义 Tensorflow 输入节点

```
a=18 #学号前两位
b=40 #学号后两位
step=((2*math.pi-b)/a+b/a)/2000 #计算步长
x=np.arange(2000.)
y=np.arange(2000.)
for i in range(0,2000):
    x[i]=-b/a+i*step
    y[i]=math.cos(a*x[i]+b)
```

#### (2) 定义“学习参数”的变量

```
w1= tf.Variable(tf.random_uniform([1]),name='w1')
w2= tf.Variable(tf.random_uniform([1]),name='w2')
w3= tf.Variable(tf.random_uniform([1]),name='w3')
b= tf.Variable(tf.random_uniform([1]),name='b')
```

#### (3) 定义“运算”

```
# 梯度下降法优化参数(学习率为 lr)
```

```
lr = 0.01
# 定义 loss: 均方差
loss = tf.reduce_mean(tf.square(y - y_))
```

#### (4) 优化函数，优化目标

```
optimizer = tf.train.GradientDescentOptimizer(lr)
# 训练的过程就是最小化这个误差值
train = optimizer.minimize(loss)
```

#### (5) 初始化所有变量

```
#初始化全局变量
init = tf.initialize_all_variables()
# 通过 session 执行上述操作
sess = tf.Session()
sess.run(init)
```

#### (6) 迭代更新参数到最优解

```
i=0
#设置 loss 大于 0.05 时继续训练
while (sess.run(loss) > 0.5*1e-1):
    #每次迭代都要最小化 Loss 函数
    sess.run(train)

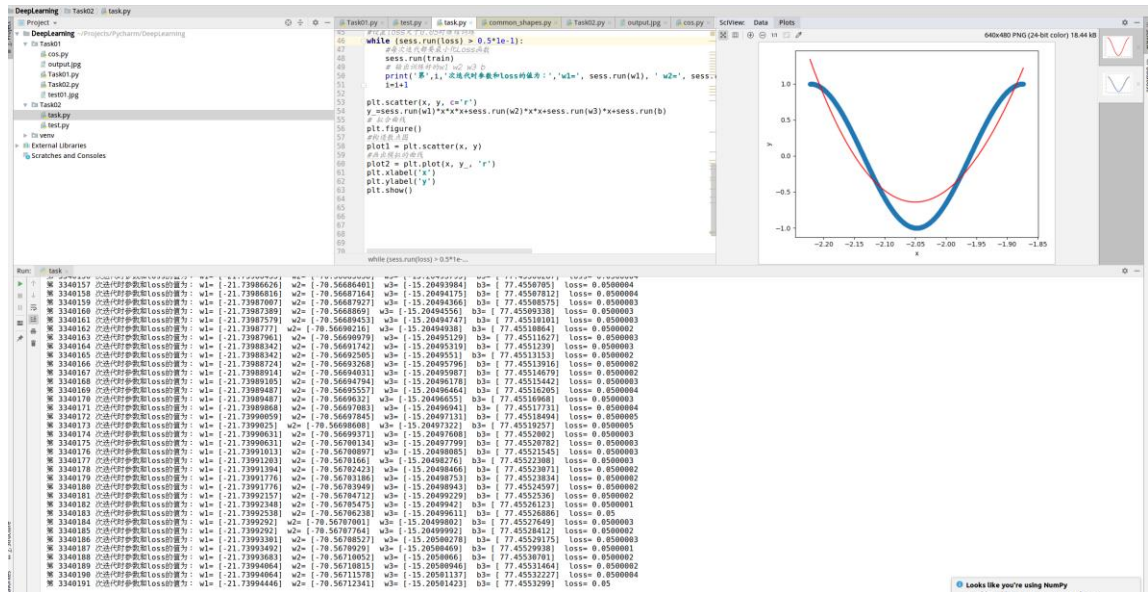
    # 输出训练好的 w1 w2 w3 b
    print('第',i,'次迭代时参数和 loss 的值为: ', 'w1=', sess.run(w1), ' w2=',
sess.run(w2), ' w3=', sess.run(w3), ' b3=', sess.run(b), '
loss=', sess.run(loss))
    i=i+1
```

#### (7) 绘制图像

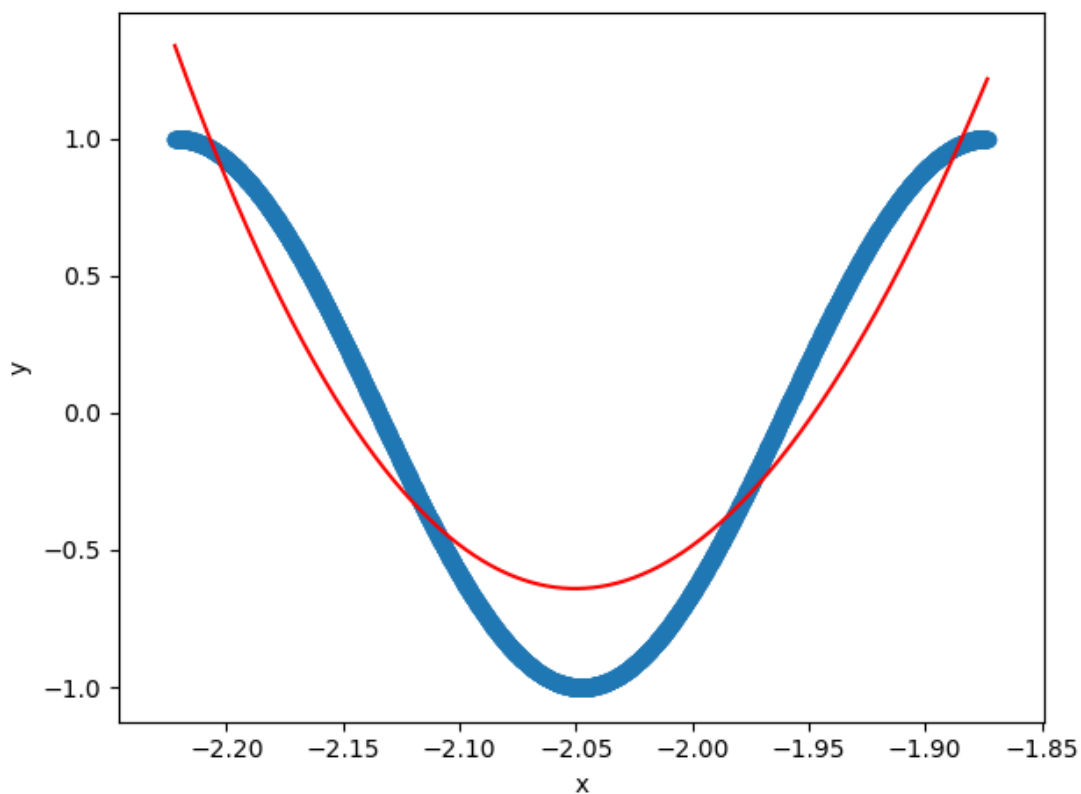
```
plt.scatter(x, y, c='r')
y_=sess.run(w1)*x*x*x+sess.run(w2)*x*x+sess.run(w3)*x+sess.run(b)
# 拟合曲线
plt.figure()
#构造散点图
plot1 = plt.scatter(x, y)
#画出模拟的曲线，红色部分
plot2 = plt.plot(x, y_, 'r')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

由于参数（学习率）和迭代终止条件的设置不同，实验结果也不同。在学习率为 0.01，迭代终止条件为 loss 小于等于 0.05 时，迭代了 3340191 次后实验

结束。图像如下：

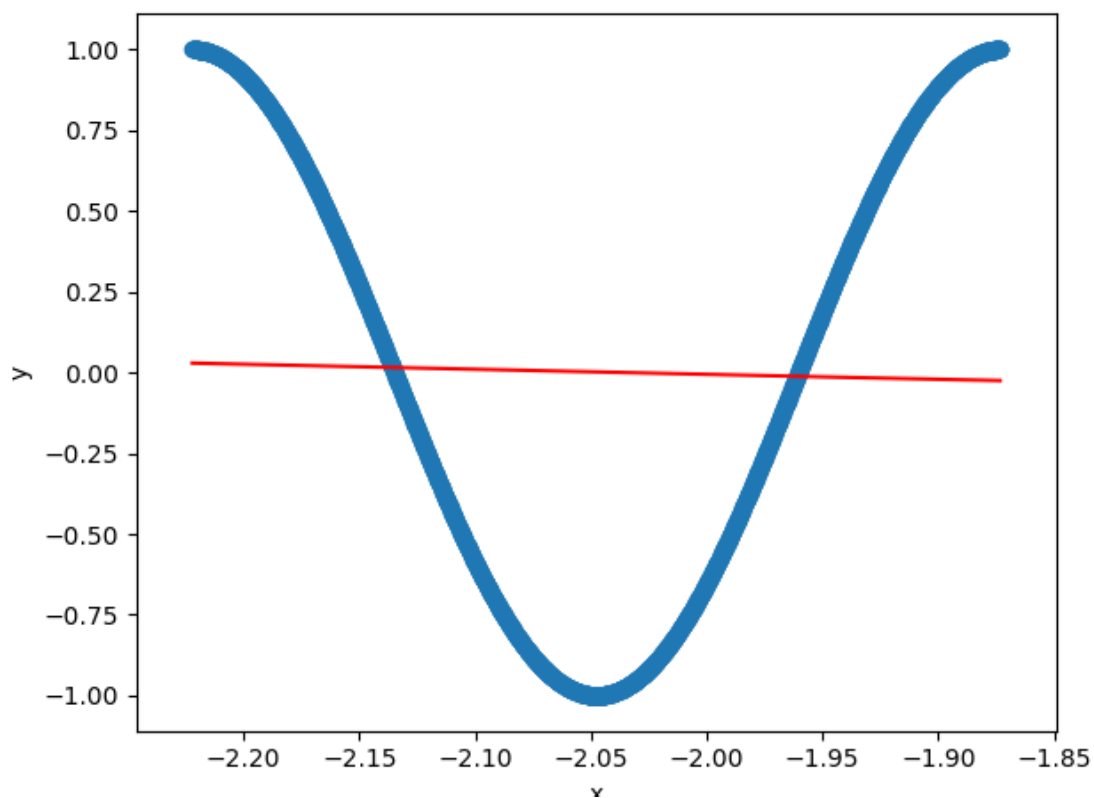


Step	Iteration	Parameter	Value	Loss
3340157	157	loss	0.0500004	0.0500004
3340158	158	w1	-21.73986626	0.0500004
3340159	159	w2	-70.56686401	0.0500004
3340160	160	w3	-15.20493984	0.0500004
3340161	161	b3	77.4550705	0.0500004
3340162	162	w1	-21.73986816	0.0500004
3340163	163	w2	-70.56687164	0.0500004
3340164	164	w3	-15.20494175	0.0500004
3340165	165	b3	77.45507812	0.0500004
3340166	166	w1	-21.73987007	0.0500003
3340167	167	w2	-70.56687927	0.0500003
3340168	168	w3	-15.20494366	0.0500003
3340169	169	b3	77.45508575	0.0500003
3340170	170	w1	-21.73987389	0.0500003
3340171	171	w2	-70.5668869	0.0500003
3340172	172	w3	-15.20494556	0.0500003
3340173	173	b3	77.45509338	0.0500003
3340174	174	w1	-21.73987579	0.0500003
3340175	175	w2	-70.56689453	0.0500003
3340176	176	w3	-15.20494747	0.0500003
3340177	177	b3	77.45510181	0.0500003
3340178	178	w1	-21.73987771	0.0500002
3340179	179	w2	-70.56690216	0.0500002
3340180	180	w3	-15.20494938	0.0500002
3340181	181	b3	77.45510864	0.0500002
3340182	182	w1	-21.73987961	0.0500002
3340183	183	w2	-70.56690979	0.0500002
3340184	184	w3	-15.20495129	0.0500002
3340185	185	b3	77.45511627	0.0500002
3340186	186	w1	-21.73988342	0.0500002
3340187	187	w2	-70.56692505	0.0500002
3340188	188	w3	-15.20495319	0.0500002
3340189	189	b3	77.45512391	0.0500002
3340190	190	w1	-21.73988541	0.0500002
3340191	191	w2	-70.56694031	0.0500002
3340192	192	w3	-15.20495511	0.0500002
3340193	193	b3	77.45513153	0.0500002
3340194	194	w1	-21.73988741	0.0500002
3340195	195	w2	-70.56695796	0.0500002
3340196	196	w3	-15.20495796	0.0500002
3340197	197	b3	77.45513916	0.0500002
3340198	198	w1	-21.73988914	0.0500002
3340199	199	w2	-70.56696431	0.0500002
3340200	200	w3	-15.20495987	0.0500002
3340201	201	b3	77.45514679	0.0500002
3340202	202	w1	-21.73989105	0.0500002
3340203	203	w2	-70.56697845	0.0500002
3340204	204	w3	-15.20496178	0.0500002
3340205	205	b3	77.45515442	0.0500002
3340206	206	w1	-21.73989291	0.0500002
3340207	207	w2	-70.56699371	0.0500002
3340208	208	w3	-15.20496464	0.0500002
3340209	209	b3	77.45516285	0.0500002
3340210	210	w1	-21.73989487	0.0500002
3340211	211	w2	-70.56699557	0.0500002
3340212	212	w3	-15.20496655	0.0500002
3340213	213	b3	77.45516968	0.0500002
3340214	214	w1	-21.73989681	0.0500002
3340215	215	w2	-70.56700897	0.0500002
3340216	216	w3	-15.20496941	0.0500002
3340217	217	b3	77.45517311	0.0500002
3340218	218	w1	-21.73989868	0.0500002
3340219	219	w2	-70.56702381	0.0500002
3340220	220	w3	-15.20497191	0.0500002
3340221	221	b3	77.45517731	0.0500002
3340222	222	w1	-21.73989925	0.0500002
3340223	223	w2	-70.56703841	0.0500002
3340224	224	w3	-15.20497596	0.0500002
3340225	225	b3	77.45518181	0.0500002
3340226	226	w1	-21.73990059	0.0500002
3340227	227	w2	-70.56705321	0.0500002
3340228	228	w3	-15.20498085	0.0500002
3340229	229	b3	77.45518649	0.0500002
3340230	230	w1	-21.73990203	0.0500002
3340231	231	w2	-70.56706868	0.0500002
3340232	232	w3	-15.20498276	0.0500002
3340233	233	b3	77.45519257	0.0500002
3340234	234	w1	-21.73990311	0.0500002
3340235	235	w2	-70.56708412	0.0500002
3340236	236	w3	-15.20498466	0.0500002
3340237	237	b3	77.45519688	0.0500002
3340238	238	w1	-21.73990421	0.0500002
3340239	239	w2	-70.56710052	0.0500002
3340240	240	w3	-15.20498753	0.0500002
3340241	241	b3	77.45520021	0.0500002
3340242	242	w1	-21.73990531	0.0500002
3340243	243	w2	-70.56711578	0.0500002
3340244	244	w3	-15.20498943	0.0500002
3340245	245	b3	77.45520384	0.0500002
3340246	246	w1	-21.73990641	0.0500002
3340247	247	w2	-70.56712341	0.0500002
3340248	248	w3	-15.20499123	0.0500002
3340249	249	b3	77.45520782	0.0500002
3340250	250	w1	-21.73990751	0.0500002
3340251	251	w2	-70.56713061	0.0500002
3340252	252	w3	-15.20499309	0.0500002
3340253	253	b3	77.45521239	0.0500002
3340254	254	w1	-21.73990861	0.0500002
3340255	255	w2	-70.56713781	0.0500002
3340256	256	w3	-15.20499423	0.0500002
3340257	257	b3	77.45521623	0.0500002
3340258	258	w1	-21.73990971	0.0500002
3340259	259	w2	-70.56714591	0.0500002
3340260	260	w3	-15.20499511	0.0500002
3340261	261	b3	77.45521975	0.0500002
3340262	262	w1	-21.73991081	0.0500002
3340263	263	w2	-70.56715301	0.0500002
3340264	264	w3	-15.20499601	0.0500002
3340265	265	b3	77.45522327	0.0500002
3340266	266	w1	-21.73991191	0.0500002
3340267	267	w2	-70.56716011	0.0500002
3340268	268	w3	-15.20499691	0.0500002
3340269	269	b3	77.45522681	0.0500002
3340270	270	w1	-21.73991301	0.0500002
3340271	271	w2	-70.56716721	0.0500002
3340272	272	w3	-15.20499781	0.0500002
3340273	273	b3	77.45523035	0.0500002
3340274	274	w1	-21.73991411	0.0500002
3340275	275	w2	-70.56717431	0.0500002
3340276	276	w3	-15.20499871	0.0500002
3340277	277	b3	77.45523389	0.0500002
3340278	278	w1	-21.73991521	0.0500002
3340279	279	w2	-70.56718141	0.0500002
3340280	280	w3	-15.20499961	0.0500002
3340281	281	b3	77.45523743	0.0500002
3340282	282	w1	-21.73991631	0.0500002
3340283	283	w2	-70.56718851	0.0500002
3340284	284	w3	-15.20500051	0.0500002
3340285	285	b3	77.45524097	0.0500002
3340286	286	w1	-21.73991741	0.0500002
3340287	287	w2	-70.56719561	0.0500002
3340288	288	w3	-15.20500141	0.0500002
3340289	289	b3	77.45524451	0.0500002
3340290	290	w1	-21.73991851	0.0500002
3340291	291	w2	-70.56720271	0.0500002
3340292	292	w3	-15.20500231	0.0500002
3340293	293	b3	77.45524805	0.0500002
3340294	294	w1	-21.73991961	0.0500002
3340295	295	w2	-70.56720981	0.0500002
3340296	296	w3	-15.20500321	0.0500002
3340297	297	b3	77.45525159	0.0500002
3340298	298	w1	-21.73992071	0.0500002
3340299	299	w2	-70.56721691	0.0500002
3340300	300	w3	-15.20500411	0.0500002
3340301	301	b3	77.45525513	0.0500002
3340302	302	w1	-21.73992181	0.0500002
3340303	303	w2	-70.56722401	0.0500002
3340304	304	w3	-15.20500501	0.0500002
3340305	305	b3	77.45525867	0.0500002
3340306	306	w1	-21.73992291	0.0500002
3340307	307	w2	-70.56723111	0.0500002
3340308	308	w3	-15.20500591	0.0500002
3340309	309	b3	77.45526221	0.0500002
3340310	310	w1	-21.73992401	0.0500002
3340311	311	w2	-70.56723821	0.0500002
3340312	312	w3	-15.20500681	0.0500002
3340313	313	b3	77.45526575	0.0500002
3340314	314	w1	-21.73992511	0.0500002
3340315	315	w2	-70.56724531	0.0500002
3340316	316	w3	-15.20500771	0.0500002
3340317	317	b3	77.45526929	0.0500002
3340318	318	w1	-21.73992621	0.0500002
3340319	319	w2	-70.56725241	0.0500002
3340320	320	w3	-15.20500861	0.0500002
3340321	321	b3	77.45527283	0.0500002
3340322	322	w1	-21.73992731	0.0500002
3340323	323	w2	-70.56725951	0.0500002
3340324	324	w3	-15.20500951	0.0500002
3340325	325	b3	77.45527637	0.0500002
3340326	326	w1	-21.73992841	0.0500002
3340327	327	w2	-70.56726661	0.0500002
3340328	328	w3	-15.20501041	0.0500002
3340329	329	b3	77.45527991	0.0500002
3340330	330	w1	-21.73992951	0.0500002
3340331	331	w2	-70.56727371	0.0500002
3340332	332	w3	-15.20501131	0.0500002
3340333	333	b3	77.45528345	0.0500002
3340334	334	w1	-21.73993061	0.0500002
3340335	335	w2	-70.56728081	0.0500002
3340336	336	w3	-15.20501221	0.0500002
3340337	337	b3	77.45528699	0.0500002
3340338	338	w1	-21.73993171	0.0500002
3340339	339	w2	-70.56728791	0.0500002
3340340	340	w3	-15.20501311	0.0500002
3340341	341	b3	77.45529053	0.0500002
3340342	342	w1	-21.73993281	0.0500002
3340343	343	w2	-70.56729501	0.0500002
3340344	344	w3	-15.20501401	0.0500002
3340345	345	b3	77.45529407	0.0500002
3340346	346	w1	-21.73993391	0.0500002
3340347	347	w2	-70.56730211	0.0500002
3340348	348	w3	-15.20501491	0.0500002
3340349	349	b3	77.45529761	0.0500002
3340350	350	w1	-21.73993501	0.0500002
3340351	351	w2	-70.56730921	0.0500002
3340352	352	w3	-15.20501581	0.0500002
3340353	353	b3	77.45530115	0.0500002
3340354	354	w1	-21.73993611	0.0500002
3340355	355	w2	-70.56731631	0.0500002
3340356	356	w3	-15.20501671	



1.1 学习率为 0.01 时的拟合图像

若学习率设置不当或迭代停止条件中的 loss 过大则会出现模拟得到的曲线近似直线的情况：（此时学习率为 0.001，终止条件为：loss 小于等于 0.5）



## 1.2 模拟曲线为直线的情况

### 1.2 ckpt 模式和 bp 模式的保存

#### (1) ckpt 模式

```
saver = tf.train.Saver()
```

```
#ckpt 模式保存
```

```
save_path = saver.save(sess, 'save_ckpt/save.ckpt')
```

#### (2) bp 模式

首先，将要保存的变量命名

```
w1= tf.Variable(tf.random_uniform([1]),name='w1')
```

```
w2= tf.Variable(tf.random_uniform([1]),name='w2')
```

```
w3= tf.Variable(tf.random_uniform([1]),name='w3')
```

```
b= tf.Variable(tf.random_uniform([1]),name='b')
```

然后，根据变量名进行保存

```
constant_graph = graph_util.convert_variables_to_constants(sess,
```

```
sess.graph_def, ["w1", "w2", "w3", "b"])
```

```
with tf.gfile.GFile('save_pb.pb', mode='wb') as f:
```

```
    f.write(constant_graph.SerializeToString())
```

保存文件末尾加入如下语句

```
#清除默认图的堆栈，并设置全局图为默认图
```

```
#解决多次运行之后的错误：NotNotFoundError (see above for traceback): Key
```

```
Variable_1 not found in checkpoint
```

```
tf.reset_default_graph()
```

### 1.3 ckpt 和 bp 模式的恢复

#### (1) ckpt 模式恢复

ckpt 多次恢复时出现的一个错误如下：

```
2019-05-15 10:25:00.038946: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 i
2019-05-15 10:25:00.038958: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 i
2019-05-15 10:25:00.038961: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX inst
2019-05-15 10:25:00.038964: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 ins
2019-05-15 10:25:00.038967: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA inst
2019-05-15 10:25:00.055361: W tensorflow/core/framework/op_kernel.cc:1191] Not found: Key Variable not found in checkpoint
2019-05-15 10:25:00.055369: W tensorflow/core/framework/op_kernel.cc:1191] Not found: Key Variable_1 not found in checkpoint
2019-05-15 10:25:00.055362: W tensorflow/core/framework/op_kernel.cc:1191] Not found: Key Variable_3 not found in checkpoint
2019-05-15 10:25:00.055378: W tensorflow/core/framework/op_kernel.cc:1191] Not found: Key Variable_2 not found in checkpoint
Traceback (most recent call last):
  File "/home/yyl/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1327, in _do_call
    return fn(*args)
  File "/home/yyl/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/client/session.py", line 1306, in _run_fn
    status, run_metadata)
  File "/home/yyl/anaconda3/envs/tensorflow/lib/python3.6/contextlib.py", line 88, in __exit__
    next(self.gen)
  File "/home/yyl/anaconda3/envs/tensorflow/lib/python3.6/site-packages/tensorflow/python/framework/errors_impl.py", line 466, in rais
    pywrap_tensorflow.TF_GetCode(status))
tensorflow.python.framework.errors_impl.NotFoundError: Key Variable not found in checkpoint
[[Node: save/RestoreV2 = RestoreV2[dtypes=[DT_FLOAT], _device="/job:localhost/replica:0/task:0/cpu:0"]](_arg_save/Const_0_0, save/
```

### 1.3 ckpt 恢复多次时出现的错误

**网上查询资料原因：**出现这样的原因是因为 ckpt 模式下第一次加载完成后，已经有名字为“w1”这样的变量，再次加载时因为名字重复，名字变为“w1\_1”，故保存的变量名已经改变了，因此出现 NotFoundError: Key Variable not found in checkpoint 类似的错。

**解决方法：**在恢复时，给变量命名并在保存模型时加上如下语句：

```
#清除默认图的堆栈，并设置全局图为默认图
tf.reset_default_graph()
```

**恢复模型参数代码：**

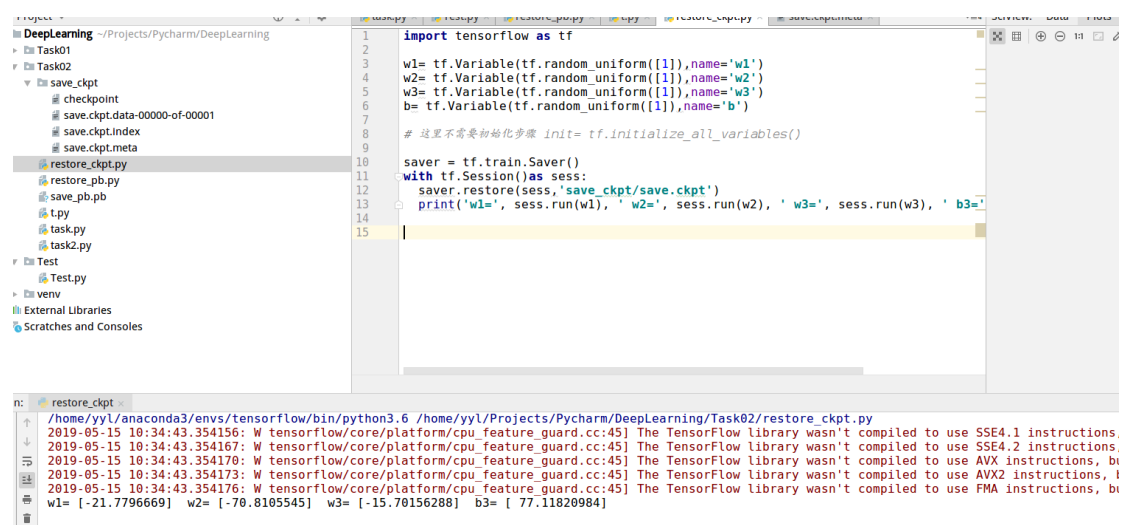
```
import tensorflow as tf

w1= tf.Variable(tf.random_uniform([1]),name='w1')
w2= tf.Variable(tf.random_uniform([1]),name='w2')
w3= tf.Variable(tf.random_uniform([1]),name='w3')
b= tf.Variable(tf.random_uniform([1]),name='b')

# 这里不需要初始化步骤 init= tf.initialize_all_variables()

saver = tf.train.Saver()
with tf.Session()as sess:
    saver.restore(sess,'save_ckpt/save.ckpt')
    print('w1=', sess.run(w1), ' w2=', sess.run(w2), ' w3=', sess.run(w3), '
b3=', sess.run(b))
```

此后，多次恢复模型也不会报错，且输出结果如下图所示：



The screenshot shows the PyCharm IDE with a project named 'DeepLearning'. The file explorer on the left shows a directory structure for 'Task02' containing files like 'save\_ckpt', 'checkpoint', 'save.ckpt.data-00000-of-00001', 'save.ckpt.index', 'save.ckpt.meta', 'restore\_ckpt.py', 'restore\_pb.py', 'save\_pb.py', 't.py', 'task.py', and 'task2.py'. The main editor displays the 'restore\_ckpt.py' script, which is identical to the code provided in the previous blocks. The console at the bottom shows the execution output for the script, including TensorFlow warnings about SSE4.1, SSE4.2, AVX, and FMA instructions, and the final variable values: w1= [-21.7796669] w2= [-70.8105545] w3= [-15.70156288] b3= [ 77.11820984].

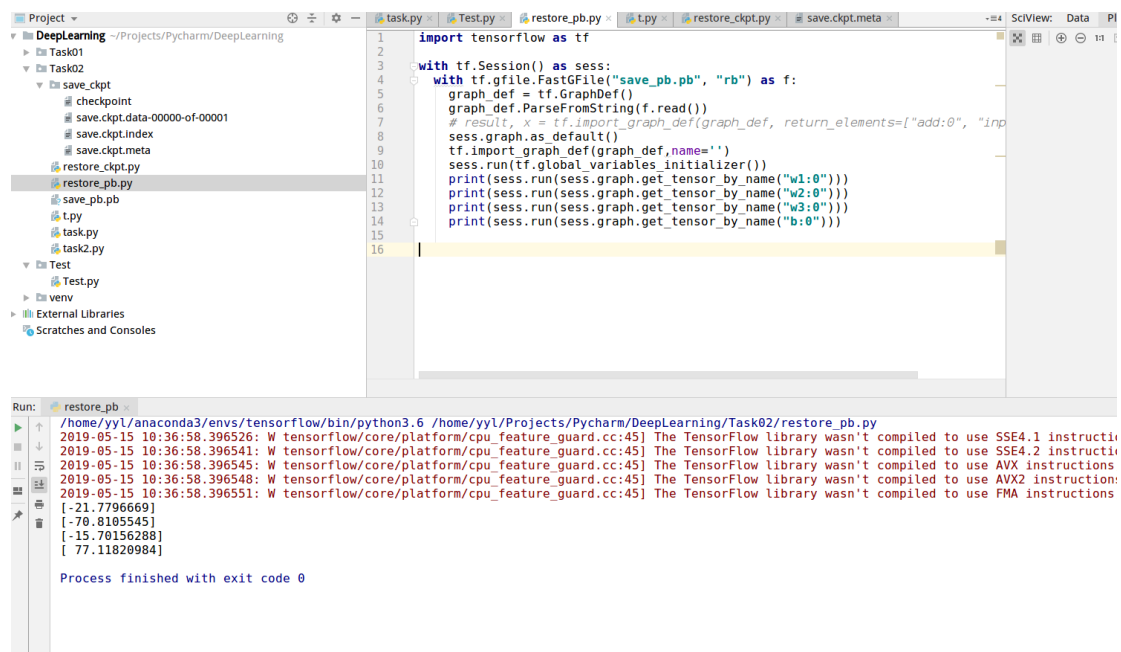
## 1.4 ckpt 恢复模型参数的结果

## (2) bp 模式恢复

```
import tensorflow as tf

with tf.Session() as sess:
    with tf.gfile.FastGFile("save_pb.pb", "rb") as f:
        graph_def = tf.GraphDef()
        graph_def.ParseFromString(f.read())
        sess.graph.as_default()
        tf.import_graph_def(graph_def, name='')
        sess.run(tf.global_variables_initializer())
        print(sess.run(sess.graph.get_tensor_by_name("w1:0")))
        print(sess.run(sess.graph.get_tensor_by_name("w2:0")))
        print(sess.run(sess.graph.get_tensor_by_name("w3:0")))
        print(sess.run(sess.graph.get_tensor_by_name("b:0")))
```

输出结果如下：



```
1 import tensorflow as tf
2
3 with tf.Session() as sess:
4     with tf.gfile.FastGFile("save_pb.pb", "rb") as f:
5         graph_def = tf.GraphDef()
6         graph_def.ParseFromString(f.read())
7         # result, x = tf.import_graph_def(graph_def, return_elements=["add:0", "input:0"])
8         sess.graph.as_default()
9         tf.import_graph_def(graph_def, name='')
10        sess.run(tf.global_variables_initializer())
11        print(sess.run(sess.graph.get_tensor_by_name("w1:0")))
12        print(sess.run(sess.graph.get_tensor_by_name("w2:0")))
13        print(sess.run(sess.graph.get_tensor_by_name("w3:0")))
14        print(sess.run(sess.graph.get_tensor_by_name("b:0")))
15
16
```

Run: restore\_pb

```
/home/yyl/anaconda3/envs/tensorflow/bin/python3.6 /home/yyl/Projects/Pycharm/DeepLearning/Task02/restore_pb.py
2019-05-15 10:36:58.396526: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions; please use a library compiled to use SSE4.1 instructions
2019-05-15 10:36:58.396541: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions; please use a library compiled to use SSE4.2 instructions
2019-05-15 10:36:58.396545: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions; please use a library compiled to use AVX instructions
2019-05-15 10:36:58.396548: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions; please use a library compiled to use AVX2 instructions
2019-05-15 10:36:58.396551: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions; please use a library compiled to use FMA instructions
[-21.7796669]
[-70.8105545]
[-15.70156288]
[ 77.11820984]

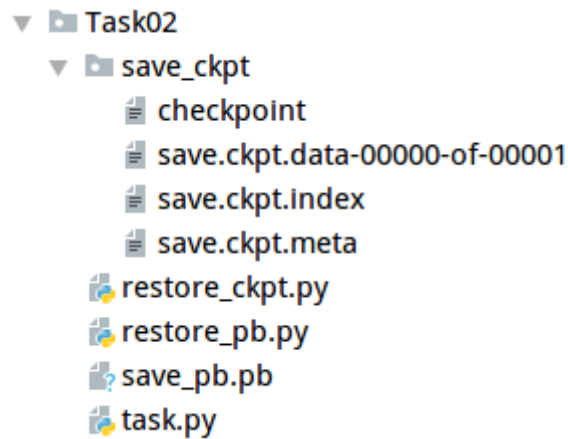
Process finished with exit code 0
```

### 1.5 bp 模式恢复模型参数的结果

可见，两种模型恢复时，参数结果一样，模型保存和恢复成功。

文件目录如下：





1.6 分别使用 ckpt 模式和 PB 模式保存拟合的模型后的文件结构

## 2. 进阶作业

使用 TensorFlow 定义函数  $y = 1 - \sin(x)/x$ ，并求解  $3y$  达到最小值时对应的  $x$

步骤提示：

- (1) 使用 TensorFlow 给出函数定义
- (2) 利用 `tf.train.GradientDescentOptimizer` 定义优化器
- (3) 启动会话，用梯度下降算法训练一定的迭代次数，在每次迭代之后输出当前的  $x$  和  $y$  值

实验代码：

```
import tensorflow as tf
import matplotlib.pyplot as plt
import numpy as np

x = tf.Variable(1, dtype = tf.float32) # 定义一个可以优化的 x 值
#定义 y 值:  $y = 1 - \sin(x)/x$ 
y = tf.subtract(1., tf.divide(tf.sin(x), x))
train_step = tf.train.GradientDescentOptimizer(0.01).minimize(y)

init = tf.global_variables_initializer()
# 生成会话，训练 1000 轮
i=0
with tf.Session() as sess:
    sess.run(init)
    for i in range(1000):
        sess.run(train_step)
        x_val = sess.run(x)
        y_val = sess.run(y)
        print("epoch", i, ":", sess.run(x), " ", sess.run(y))
    i=i+1
```

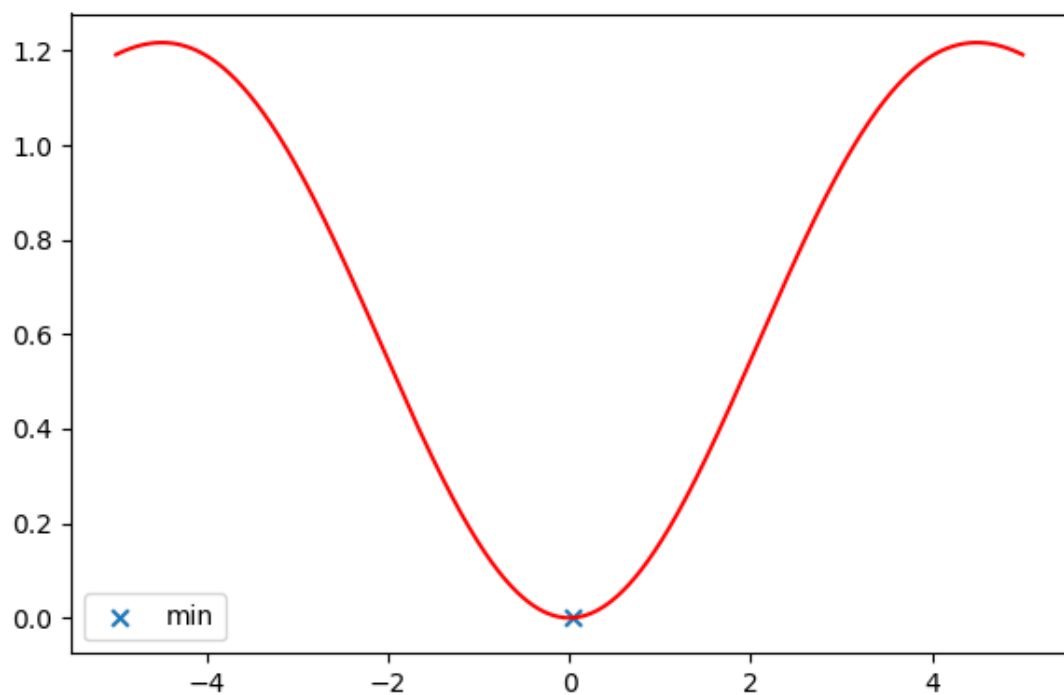


```
plt.figure(figsize=(6, 4)) # 设置图片大小
x_data=np.linspace(-5,5,10000) #这个表示在-5 到 5 之间生成 10000 个 x 值
plt.plot(x_data,1-(np.sin(x_data)/x_data),color='red',)
plt.scatter(x_val, y_val, marker = 'x', s = 40 ,label = 'min') #画出最优值
plt.legend() # 显示图例
plt.show()
```

输出：

```
epoch 962 : 0.0422763 0.000297844
epoch 963 : 0.0421354 0.000295937
epoch 964 : 0.041995 0.000293911
epoch 965 : 0.0418551 0.000291944
epoch 966 : 0.0417156 0.000289977
epoch 967 : 0.0415765 0.000288069
epoch 968 : 0.041438 0.000286162
epoch 969 : 0.0412999 0.000284195
epoch 970 : 0.0411622 0.000282347
epoch 971 : 0.041025 0.000280499
epoch 972 : 0.0408883 0.000278592
epoch 973 : 0.040752 0.000276804
epoch 974 : 0.0406162 0.000274897
epoch 975 : 0.0404808 0.000273108
epoch 976 : 0.0403459 0.000271261
epoch 977 : 0.0402114 0.000269473
epoch 978 : 0.0400774 0.000267684
epoch 979 : 0.0399438 0.000265896
epoch 980 : 0.0398107 0.000264168
epoch 981 : 0.039678 0.000262439
epoch 982 : 0.0395458 0.000260651
epoch 983 : 0.039414 0.000258863
epoch 984 : 0.0392826 0.000257194
epoch 985 : 0.0391517 0.000255466
epoch 986 : 0.0390213 0.000253737
epoch 987 : 0.0388912 0.000252128
epoch 988 : 0.0387616 0.000250341
epoch 989 : 0.0386324 0.000248671
epoch 990 : 0.0385036 0.000247121
epoch 991 : 0.0383753 0.000245392
epoch 992 : 0.0382474 0.000243783
epoch 993 : 0.0381199 0.000242174
epoch 994 : 0.0379929 0.000240505
epoch 995 : 0.0378663 0.000238955
epoch 996 : 0.03774 0.000237405
epoch 997 : 0.0376142 0.000235796
epoch 998 : 0.0374889 0.000234187
epoch 999 : 0.0373639 0.000232697
```

## 2.1 实验输出结果



2.2 实验图像

上图红色部分绘制了函数  $y = 1 - \sin(x)/x$  的图像，其中蓝色点 x 即为使用梯度下降求得的最优值  $(0.0373639, 0.000232697)$ 。