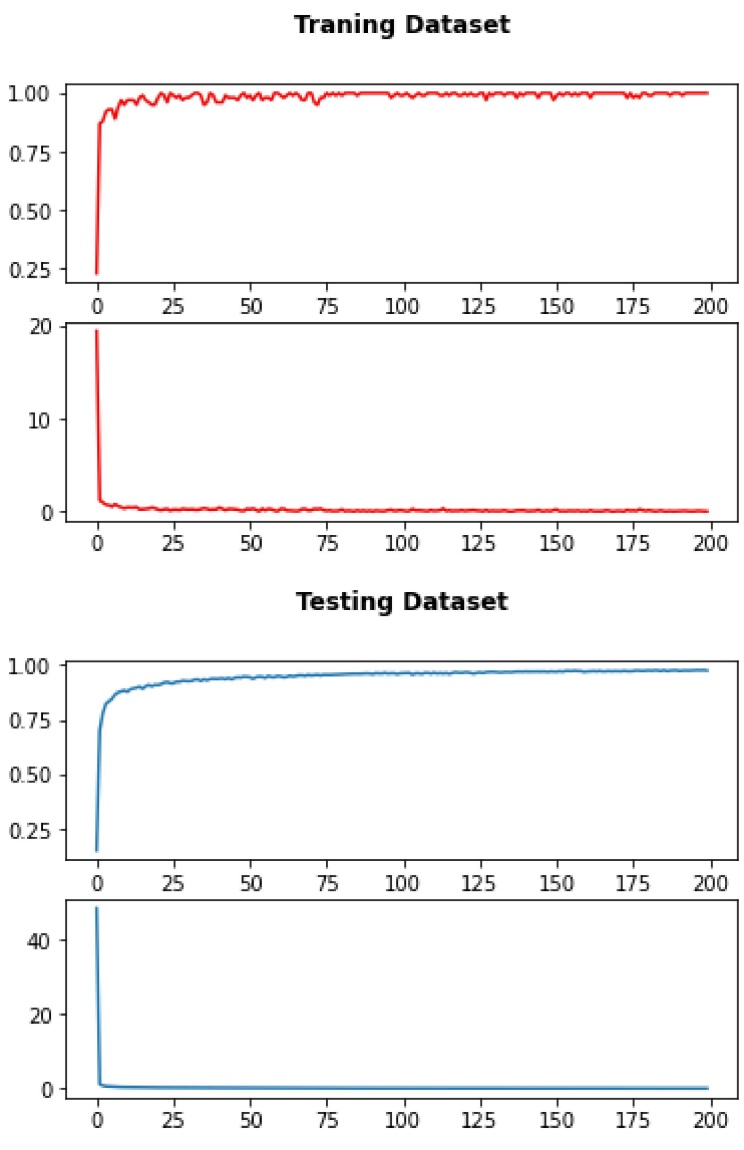
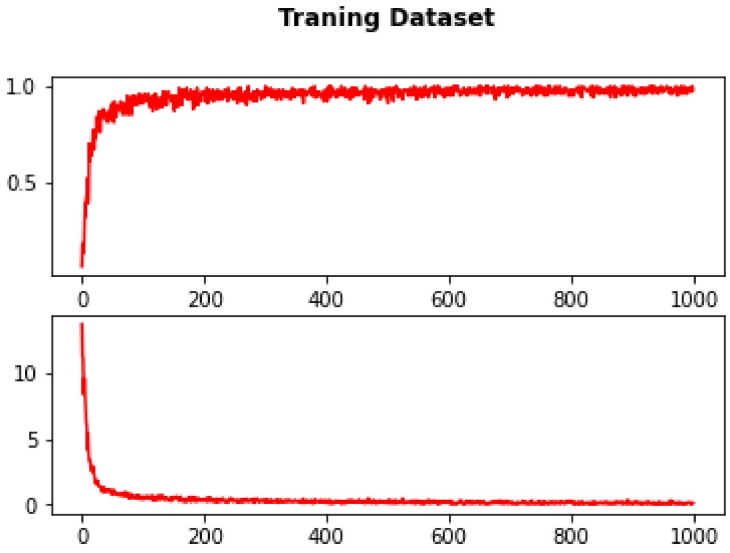
一、 多层感知机用于 MINSTT手写数字数据集分类（提交实现步骤描述以及下面要求提交的结果）

```
step : 0, training loss = 19.4287
step : 0, testing loss = 48.6643
step : 0, training accuracy = 0.23
step : 0, testing accuracy = 0.1506
step : 500, training loss = 0.241689
step : 500, testing loss = 0.173039
step : 500, training accuracy = 0.99
step : 500, testing accuracy = 0.9455
step : 1000, training loss = 0.106698
step : 1000, testing loss = 0.114709
step : 1000, training accuracy = 0.99
step : 1000, testing accuracy = 0.9641
step : 1500, training loss = 0.223329
step : 1500, testing loss = 0.0964008
step : 1500, training accuracy = 0.99
step : 1500, testing accuracy = 0.9703
```



二、 卷积神经网络用于 MINISTT手写数字数据集分类（提交实现步骤描述以及下面要求提交的结果）

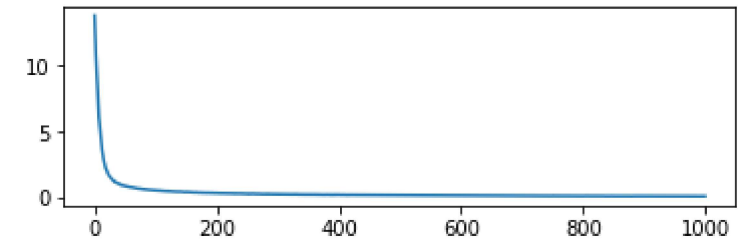
```
WARNING:tensorflow:From <ipython-input-2-d3efd2571d0a>:37: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/python/util/tf_should_use.py:198: initialize_all_variables (from tensorflow.python.ops.variables) is deprecated and will be removed after 2017-03-02.
Instructions for updating:
Use `tf.global_variables_initializer` instead.
0.0686
0.9315
0.9554
0.9631
0.9705
0.9746
0.9782
0.9825
0.9841
0.9881
```



三、 多层感知机实现异或运算（提交实现步骤描述、源代码以及最后的测试误差）

In [19]:
```
X=np.array([[0,0],[1,0],[0,1],[1,1],[0,0],[0,0],[1,1],[0,1],[1,0],[1,0],[1,1],[0,1],[0,0]])
y=np.array([[0],[1],[1],[0],[0],[0],[0],[1],[1],[1],[0],[1],[0]])
shape=[2,2]
bp_nn=net(lr=0.1,shape=shape,X=X,y=y)
for i in range(10000):
  bp_nn.feed_batch(X=X,y=y,batch_size=len(y))
bp_nn.plot_loss()
```

```
loss: 0.22050371248386125
loss: 0.14364857786783874
loss: 0.024413688022166775
loss: 0.007919176614246664
loss: 0.004416890330842479
loss: 0.003001697102667052
loss: 0.002252947662052557
loss: 0.0017942987253716863
loss: 0.0014862825745014083
loss: 0.001265954029455774
```



In [20]:
```
bp_nn.putin([0,1],[1])
print("[0,1] prediction:",bp_nn.run())
```

```
[0,1] prediction: [[0.96467548]]
```

In [21]:
```
bp_nn.putin([1,0],[1])
print("[1,0] prediction:",bp_nn.run())
```

```
[1,0] prediction: [[0.96444233]]
```

In [22]:
```
bp_nn.putin([0,0],[0])
print("[0,0] prediction:",bp_nn.run())
```

```
[0,0] prediction: [[0.03407978]]
```

In [23]:
```
bp_nn.putin([1,1],[0])
print("[1,1] prediction:",bp_nn.run())
```

[1,1] prediction: [[0.03776495]]