chap4 simple neural network 报告

函数定义

我们首先定义了一个正弦函数作为目标函数，即 `true_function(x)`，这个函数将被用来生成训练集和测试集，以及评估模型的拟合效果。

```
pythonCopy codedef true_function(x):
    return tf.math.sin(x)
```

数据采集

我们使用 `np.random.uniform()` 函数在给定的范围内采样数据，并将数据转换成2维形状以适配神经网络模型。训练集包含1000个样本，测试集包含200个样本。

```
pythonCopy codetrain_size = 1000
test_size = 200

train_x = np.random.uniform(-5, 5, train_size).astype(np.float32).reshape(-1, 1)
test_x = np.random.uniform(-5, 5, test_size).astype(np.float32).reshape(-1, 1)

train_y = true_function(train_x)
test_y = true_function(test_x)
```

模型描述

我们定义了一个两层的神经网络模型 `myModel`，该模型包含一个输入层、一个隐藏层和一个输出层。隐藏层使用ReLU作为激活函数，输出层不使用激活函数。模型的损失函数使用均方误差。

```
pythonCopy codeclass myModel:
    def __init__(self):
        self.w1 = tf.Variable(tf.random.normal([1, 128]), name='w1')
        self.b1 = tf.Variable(tf.zeros([128]), name='b1')
        self.w2 = tf.Variable(tf.random.normal([128, 1]), name='w2')
        self.b2 = tf.Variable(tf.zeros([1]), name='b2')

    def __call__(self, x):
        h1 = tf.nn.relu(tf.matmul(x, self.w1) + self.b1)
        y_pred = tf.matmul(h1, self.w2) + self.b2
        return y_pred
```

拟合效果

我们使用训练集来训练模型，并使用测试集来验证模型的拟合效果。我们设置了6000个epoch来训练模型，并每隔1000个epoch输出一次损失值。最终打印出测试集上的损失值。

```
pythonCopy codefor epoch in range(6000):
    loss = train_one_step(model, optimizer, tf.constant(train_x, dtype=tf.float32),
tf.constant(train_y, dtype=tf.float32))
    if epoch % 1000 == 0:
        print('epoch', epoch, ': loss', loss.numpy())

test_loss = test(model, tf.constant(test_x, dtype=tf.float32), tf.constant(test_y,
dtype=tf.float32))
print('test loss', test_loss.numpy())
```

结果如下

```
epoch 0 : loss 418.22836
epoch 1000 : loss 0.1077287
epoch 2000 : loss 0.05339888
epoch 3000 : loss 0.02557661
epoch 4000 : loss 0.0171903
epoch 5000 : loss 0.013338562
test loss 0.01406639
```

作图查看拟合效果如下：