

函数拟合——报告

2151025 韩畅

问题描述：

理论和实验证明，一个两层的ReLU网络可以模拟任何函数[1~5]。请自行定义一个函数, 并使用基于ReLU的神经网络来拟合此函数。

要求：

- 请自行在函数上采样生成训练集和测试集，使用训练集来训练神经网络，使用测试集来验证拟合效果。
- 可以使用深度学习框架来编写模型，如tensorflow、pytorch、keras等。
 - 如果不使用上述框架，直接用NumPy实现可以最高加5分的附加分。
- 提交时请一并提交代码和报告。
 - 代码建议注释清楚（5分）
 - 报告至少应包含以下部分：（5分）
 - 函数定义、数据采集、模型描述、拟合效果。

函数定义：

```
# 定义函数
def f(x):
    return x**3 + 0.5*x**2 - 2*x + 1
```

数据采集：

```
# 采样数据
x_train = tf.random.uniform([1000, 1], minval=-2, maxval=2)
y_train = f(x_train)
x_test = tf.random.uniform([10, 1], minval=-2, maxval=2)
y_test = f(x_test)
```

模型描述：

```
# 定义模型
model = tf.keras.Sequential([
    tf.keras.layers.Dense(100, activation='relu'),
    tf.keras.layers.Dense(1)
])

# 编译模型
model.compile(optimizer='adam', loss='mse')

# 训练模型
model.fit(x_train, y_train, epochs=100)
```

使用了一个两层的ReLU网络，其中第一层有 100 个神经元，第二层有 1 个神经元。使用 Adam 优化器和均方误差损失函数来训练模型。

拟合效果：

```
1/1 — 0s 89ms/step - loss: 0.0058
测试集误差: 0.005768279079347849
```

