

一、循环神经网络用于预测任务（提交实现步骤描述以及下面要求提交的结果）

1、构建标准循环神经网络进行预测任务

(1) 数据生成：

使用函数 $y = \sin(0.06x) + U(-0.1, 0.1)$, $x = 1, 2, \dots, 600$ 生成 600 个样本点, U 为均匀分布，可视化样本。（提交可视化结果）

训练集样本点：前面 $600 \times 0.7 = 420$ 个样本点

测试集样本点：后面 $600 \times 0.3 = 180$ 个样本点

将循环神经网络的时间步长设置为 7，那么训练集大小为： $420 - 7 = 413$ ；测试集大小为： $180 - 7 = 173$ 。

例子：

输入[1, 2, 3, 4, 5, 6, 7]时刻的真实值，预测第[8]时刻的值

输入[2, 3, 4, 5, 6, 7, 8]时刻的真实值，预测第[9]时刻的值

(2) 模型架构：构建时间步长为 7，隐层单元数为 32 的循环神经网络，取最后一个时间步的隐层状态作为输出层的输入。输出层使用一个全连接层：

$$prediction = last_state * w + b$$

其中 w 为 32×1 ， b 的维度为 1。

(3) 画出训练误差随迭代次数的变化图以及测试集的预测值和真实值的对比图。

（提交最终的测试误差，变化图以及对比图）

2、构建 LSTM 循环神经网络进行预测任务（要求同上）

二、基于作业一中的 LSTM 循环神经网络应用 Dropout 正则化方法

在输出层的输入添加 Dropout 层，进行两次试验，Dropout 比率分别设为 0.2 和 0.4，画出训练误差随迭代次数的变化图以及测试集的预测值和真实值的对比图。

（提交最终的测试误差，变化图以及对比图）

三、基于作业一中的 LSTM 循环神经网络应用不同的优化方法（提交最终的测试误差，变化图以及对比图）

- 1、使用随机梯度下降算法（`tf.train.GradientDescentOptimizer`）。
- 2、使用动量梯度下降算法（`tf.train.MomentumOptimizer`）。
- 3、使用 Adam 优化算法（`tf.train.AdamOptimizer`）。