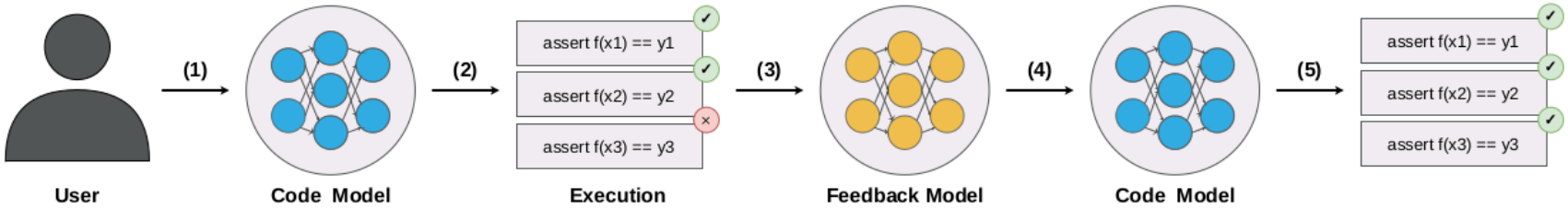
组会汇报

陈钶杰
专业:计算数学

July 11, 2023

目录

什么是语言模型的自我修复能力?



(1)

Given is a string s representing the day of the week today. s is one of SUN, MON, TUE, WED, THU, FRI, or SAT. After how many days is the next Sunday (tomorrow or later)?

```
# UNIT TESTS
# (EXECUTABLE)
assert f('MON') == 6
assert f('WED') == 4
assert f('SUN') == 7
```

(2)

```
def f(s):
  return (7 - ['SUN', ... , 'FRI', 'SAT'].index(s)) % 7
```

(3)

Given input 'SUN', the program returned 0, but the expected output was 7.

(4)

The code does not account for the case where the input is 'SUN' and the output should be 7. This can be fixed by removing the modulo operation.

(5)

```
def f(s):
  return (7 - ['SUN', ... , 'FRI', 'SAT'].index(s)) # % 7
```

- 如上图所示,是一个代码任务的自我修复能力的展示,用户给定一个任务,然后返回一个代码,之后检测该代码是否有bug,有的话进行使用反馈模型进行修复,重新检测代码,直至没有问题再输出.

模型使用的方法

- 该论文提出了一种名为pass @ t 的新评估策略，该策略根据从模型中采样的tokens总数来衡量任务的通过率，从而可以与纯粹基于采样的方法进行公平比较。
- 本文分析了GPT-3.5 和GPT-4 对包含各种编码挑战的具有挑战性的数据集进行自我修复的能力 。
- 这个的方法主要分为4个步骤,代码生成,代码执行,反馈生成,代码修复

结论

1. GPT-3.5 无法对具有挑战性的编码任务进行自我修复。
2. 尽管在GPT-4 中可以看到绩效提升，但效果不大，依赖于在初始项目中实现足够的多样性。
3. 用经验丰富的程序员提供的反馈取代GPT-4 自行生成的反馈，使通过所有单元测试的修复程序数量增加了57%。

相关启示

- 修改当前的评估指标,改成pass @ t的评估指标,使得比较更加的公平

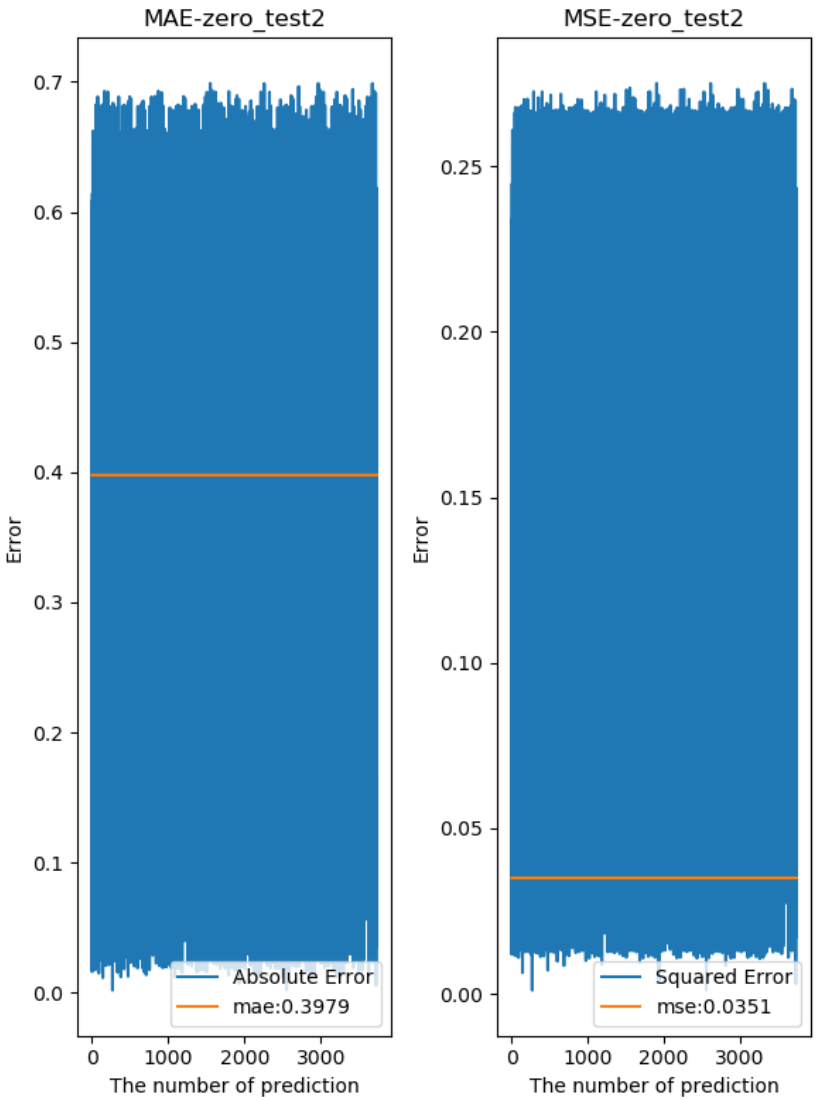
训练总数据量:30000;预测长度:16预测8;用于训练函数序列:[x, $Cos(x)$, $Sin(x^2+2)$, $Sin(x)$]

待预测的数据:8预测4;预测的函数序列:[x, $Cos(x)$, $Sin(x^2+2)$, $Sin(x)$];准确率:0%

- 最终预测的所有结果都是长度为8的,但对于单一的任务是没有泛化能力的.

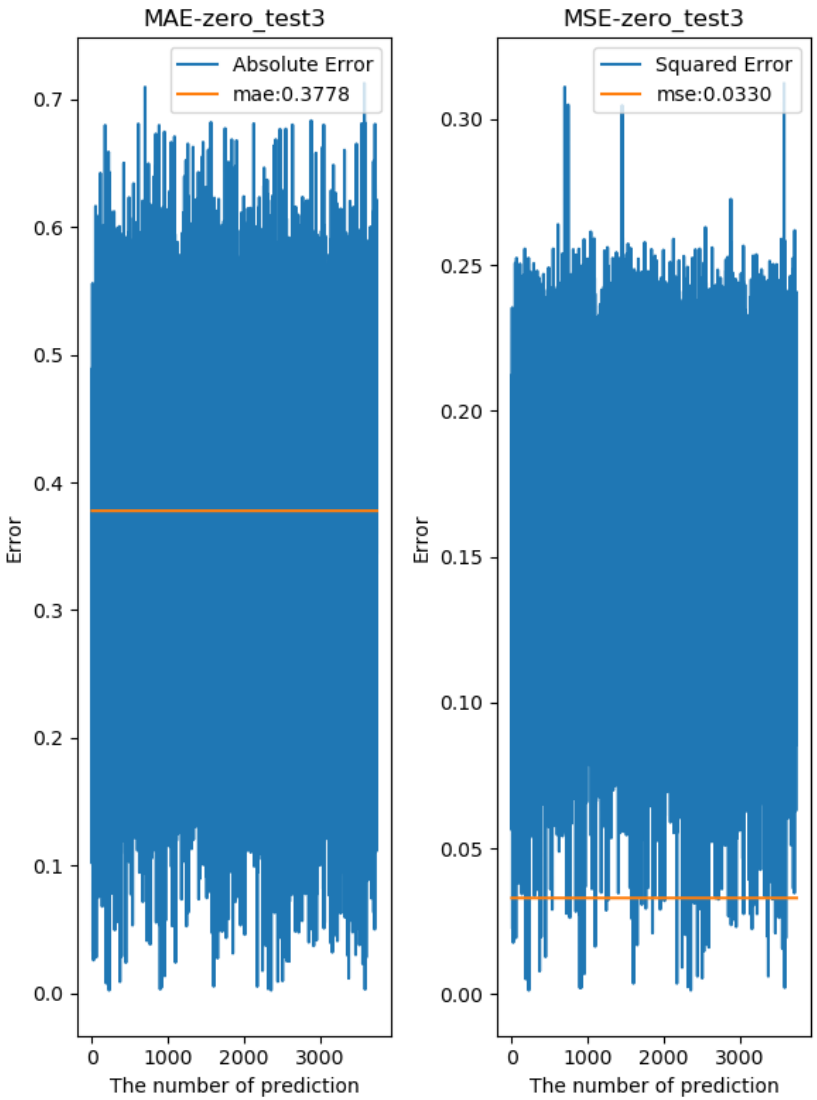
训练总数据量:30000,预测长度:16预测8,用于训练函数序列:[x, $Cos(x)$, $Sin(x^2+2)$, $Sin(x)$]

待预测的数据:16预测8,预测的函数序列:[x, $sin(x)$], 准确率:99.78%



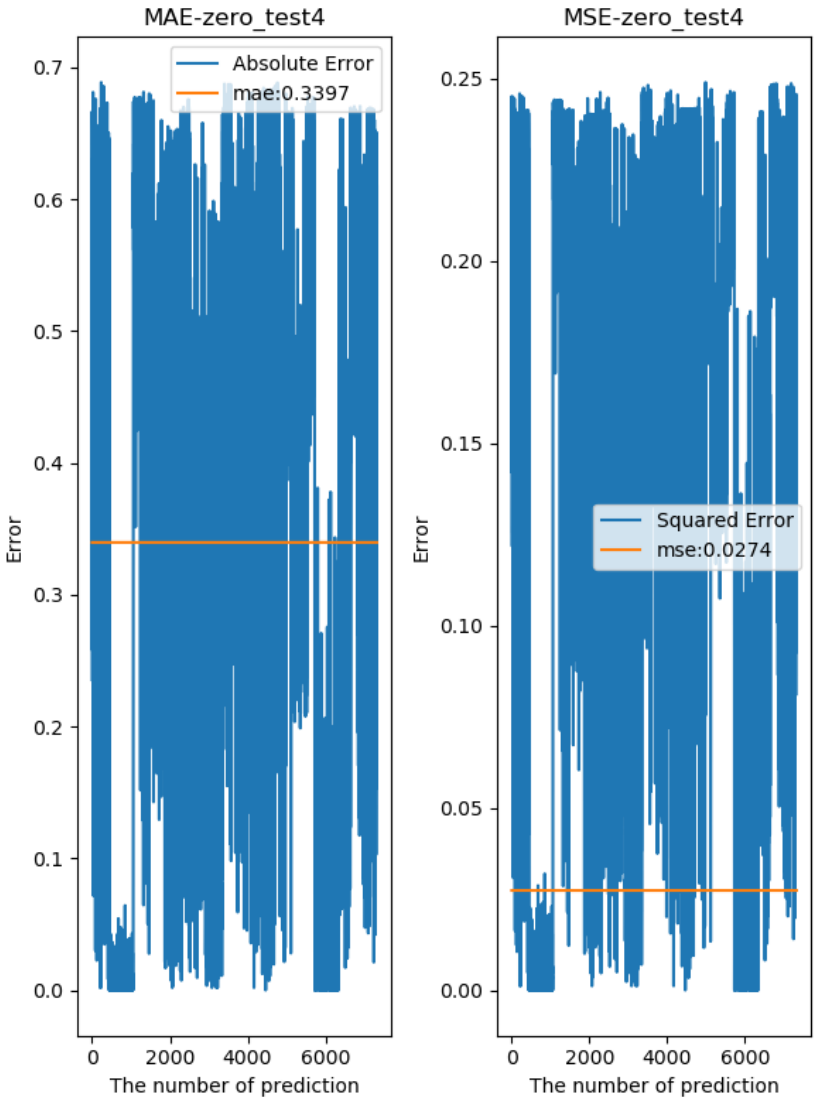
训练总数据量:30000,预测长度:16预测8,用于训练函数序列:[x, $sin(x)$]

待预测的数据:16预测8,预测的函数序列:[x, $Cos(x)$, $Sin(x^2+2)$, $Sin(x)$], 准确率:99.97%



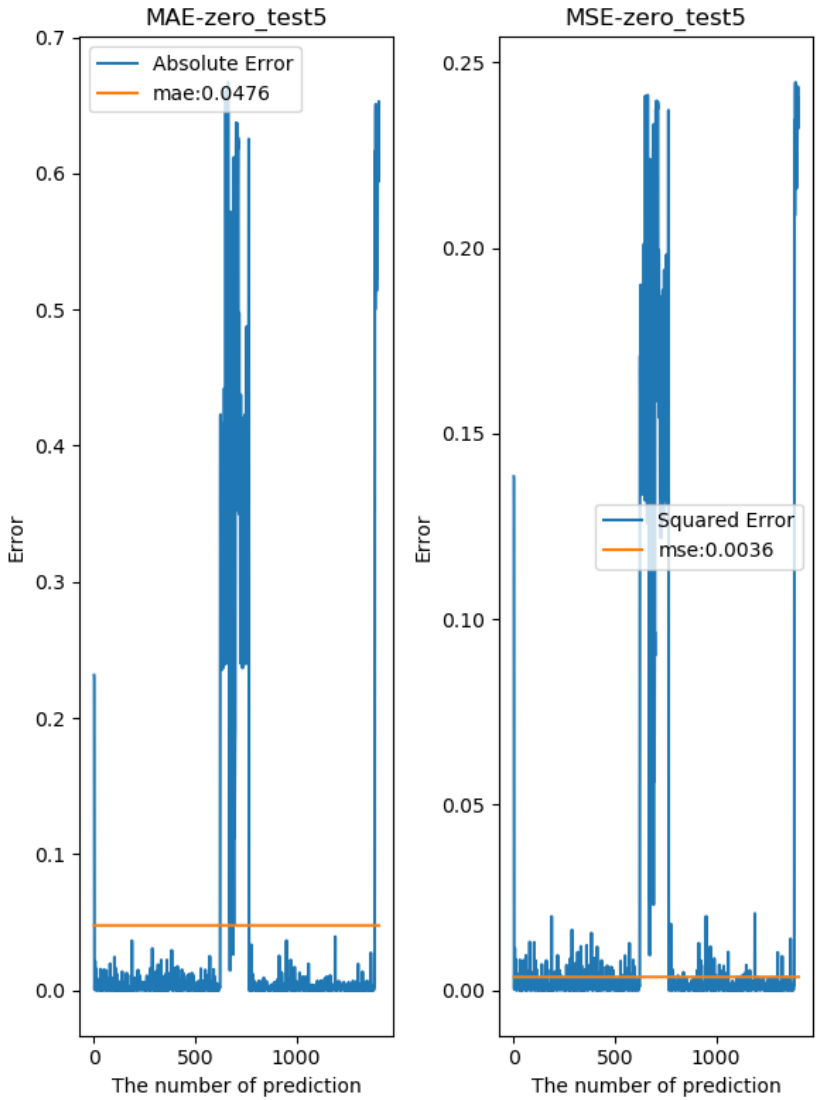
训练总数据量:100000,预测长度:8预测4,...,18预测9,其中间隔为2;用于训练的函数序列: 4种不同的函数序列;

待预测的数据:16预测8,...,80预测40,间隔为4;预测的函数序列:2种不同的函数序列;准确率:37.11%



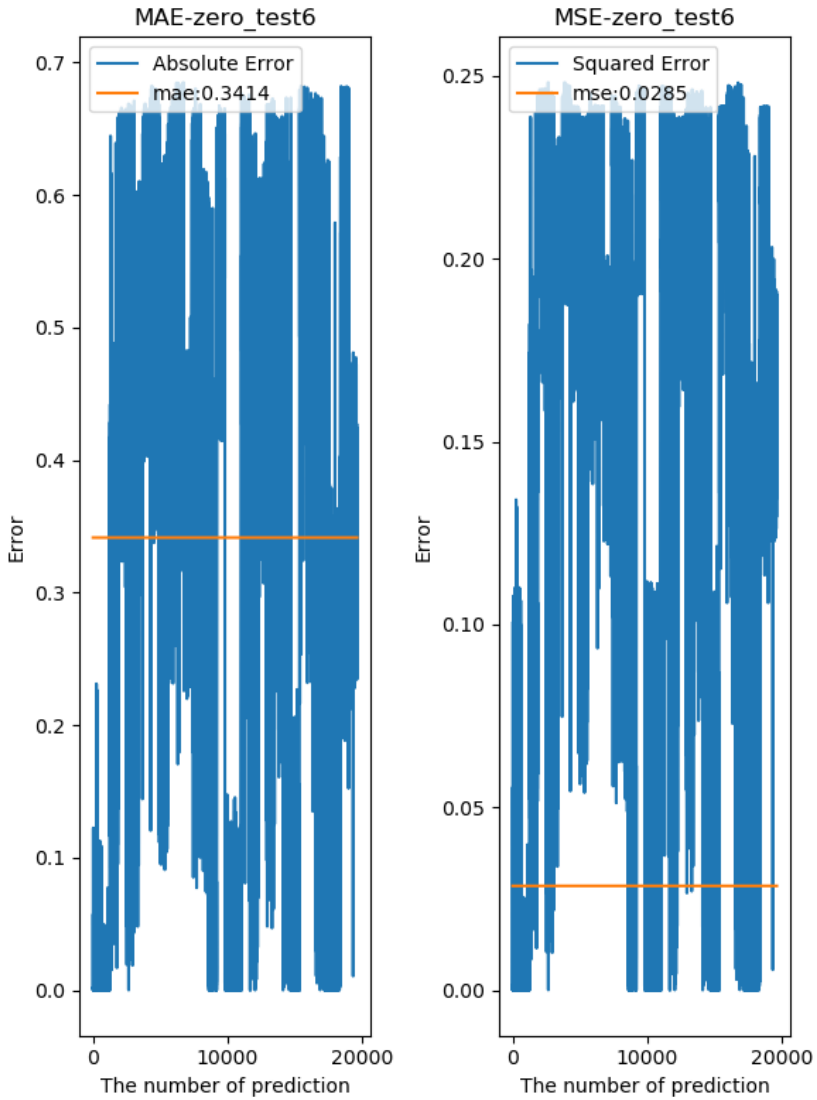
训练总数据量:120000,预测长度:8预测4,16预测8;用于训练的函数序列: 2种不同的函数序列;

待预测的数据:16预测8,...,80预测40,间隔为4;预测的函数序列:2种不同的函数序列(和训练的函数序列相同);准确率:7.14%



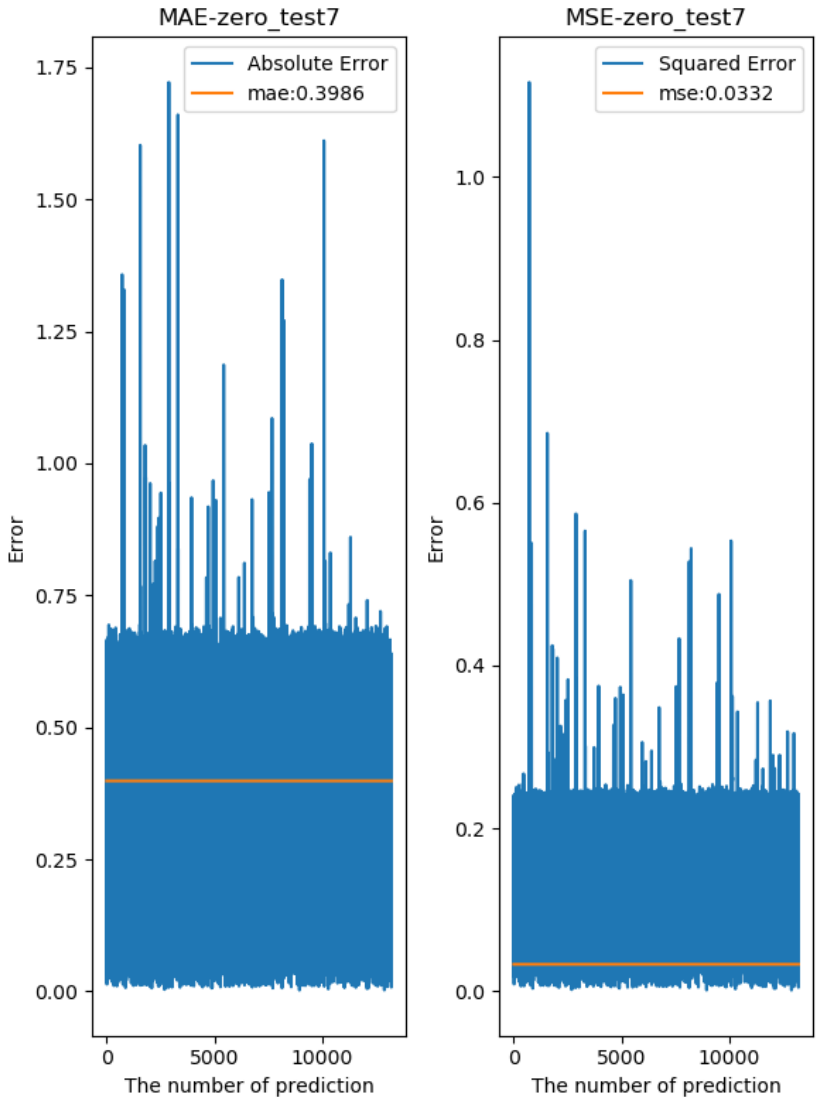
训练总数据量:160000,预测长度:8预测4,...,24预测12,间隔4;用于训练的函数序列:4种不同函数序列

待预测的数据:16预测8,...,80预测40,间隔为4;预测的函数序列:2种不同的函数序列;准确率:99.48%



训练总数据量:640000,预测长度:8预测4,...,80预测40,间隔4;用于训练的函数序列:8种基本函数,如三角,多项式函数,对数,平方根等

待预测的数据:16预测8,...,80预测40,间隔为4;预测的函数序列:2种不同的函数序列;准确率:66.95%



实验结论

- 第一个实验中,预测的函数序列和训练的函数序列相同,但预测序列的长度和训练的长度不同,,结果发现正确的预测长度应该是4,但是结果预测长度全部是8.从中可以看出,模型对于解决预测长度不同的问题的能力较弱.
- 第二个实验中,训练的函数不同,但是预测和训练的长度都相同,准确率明显上升,从中可知
  1. 与第一个实验相比,想要模型有较高的准确率,需要和训练数据的长度尽可能的接近.
  2. 在没见过的函数序列预测上,相比预测同一个函数序列,误差明显变大.之前的实验中MAE是0.001
- 第三个实验中,训练的函数不同,但是预测和训练的长度都相同,相比上一个实验中,这个实验测试的是维度更高的函数序列预测维度更低的函数序列是否有更高的精度.
  1.
  2. 从结果的准确率和最终预测误差来看,使用更高维度来预测低维度和低维度预测高维度的基本差不多,所以函数维度大小对模型泛化性能影响不大.

实验结论

- 第四个实验中,训练和预测的序列长度和种类都不同.即去预测一个长度,函数序列都不同的序列,有如下的结论
  1. 准确率较低,预测结果的误差也比较大.并但是训练的模型更复杂,相比第一种实验中训练简单的模型,准确率上有明显的提升.
- 第五个实验中,相比第四个实验中,将序列数量和函数种类都减少,再去预测相同的内容,最终的发现准确率更低了.
  1. 这说明训练时更复杂的数据集,能够让准确率提高.
  2. 预测结果误差由于样本少,不好判断
- 第六个实验中,预测长度更接近和且更多种类的函数序列来进行预测,且将单种类型的数据量从之前实验的5000提高到10000.
  1. 从结果中可以看到,数据量提高且训练的序列长度和预测的长度更接近以后,最终的准确率会有显著提高.
  2. 预测结果的误差依旧没有明显的减小.

实验结论

- 第七个实验中,用了一个预测序列形式相同,但是函数序列种类更多的一个模型,对相同的数据进行预测.得到结果
  1. 从准确率中可以看到,当预测和训练的长度相同时,训练的函数序列种类变多反而导致最终的准确率下降了.
  2. 预测结果的误差也没有发生明显的变化.

实现目标

- 修改提示方式,比如表示数字的时候用$符号等方式。
- 如何提高模型的泛化能力,在准确率上可以让训练的模型尽可能的进行多种不同长度序列的测试.在预测结果的误差上,还不太清楚如何减小误差.
- 现在最大的问题就是,如何能够改善模型的泛化能力,特别是在预测结果误差上的改进.

谢谢老师和同学的聆听!