关键应该在loss计算上,我试了mask,插入sep,加注意力,几乎都不能学习。最后改损失函数为0-1并重写了grad-function才勉强有点预测效果,但效果还是很差。

其他可能影响训练效果的因素:

- 1 表达式的 ASM代码长度参差不齐,输入和输出的前几位得到的训练次数远比后面几位的训练次数多,因此数据可能需要强化
- 2最开始将表达式根节点固定到输入第一位,剩下节点根据与根节点的距离在输入与label中后延。原本认为此方法可能有助于网络学习,结果发现此方法对训练有害

这是2*\$x^2+10*x+(1+2)==0 对应的embedding的内容 (input 与label) ,下方为对应ASM解析所得内容,亦即embedding前内容(部分内容未导出),详细每个element对应含义请见 excle文档

"grounded": false,
"extendedOperand1": {
 "argName": "x",

"argType": "identifier",

"changeable": 1,
"className": "".

```
预测最多20行代码,分多批输入
   "extended_code_segments": "[[
 ASM的结束,此行对应方程中的x^2->...)
     0.0,\ 0.0,\ 1.0,\ 1.0,\ 1.0,\ 0.0,\ 2.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 1.0,\ 0.0,\ 1.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 1.0,\ 0.0,\ 1.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,
0.0,\ 0.0,\ 1.0,\ 1.0,\ 0.0,\ 1.0,\ 0.0,\ 1.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,
 (-13用于标识一段ASM的结束,此行ASM对应方程中的 ...==0, 也是label对应的行)
-11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -11.0, -1
 1.0, -11.0
     (-11是padding, 会被mask掉) ]]"
 "extended ounding state": {
                                                                                                    "extendedCodeTable": {
                                                                                                                                       "extendedCodeList": [
                                                                                                                                                                                                                                                                                                                                                     ASM代码段
```

```
arglype : identifier ,
    "changeable": 1,
    "className": "",
    "isClass": 0,
    "occurrenceNumber": 1.
    "isLeafNode": true,
                            是否是叶节点
    "positionInAST": [
                          在AST中相对root的位置,左跳-1,右跳1
        -1,
        -1,
        -1,
        1.
        ---1
    "argValue": null
"extendedOperand2": {
    "argName": null,
    "argType": "number",
    "changeable": 0,
    "className": "",
    "isClass": 0,
    "occurrenceNumber": 1,
    "isLeafNode": true,
    "positionInAST": [
        -1,
        -1.
    "argValue": 2.0
"extendedOperator": {
   "argName": "^",
    "argType": "other"
"extendedResult": {
    "argName": "varLLLR",
    "argType": "identifier",
    "changeable": 1,
    "className": "",
    "isClass": 0,
    "occurrenceNumber": 0,
    "isLeafNode": false,
    "positionInAST": [
        -1,
        -1.
        -1,
        1
    "argValue": null
```

```
"grounded": false,
"extendedOperand1": {
    "argName": null,
    "argType": "number",
    "changeable": 0,
    "className": "",
    "isClass": 0,
    "occurrenceNumber": 1,
    "isLeafNode": true,
    "positionInAST": [
        -1.
        -1_{i}
        -1,
        --1
    "argValue": 2.0
"extendedOperand2": {
    "argName": "varLLLR",
    "argType": "identifier",
    "changeable": 1,
    "className": "",
    "isClass": 0,
    "occurrenceNumber": 1,
    "isLeafNode": false,
    "positionInAST": [
        -1,
        -1.
        -1,
    "argValue": null
"extendedOperator": {
    "argName": "*",
    "argType": "other"
"extendedResult": {
    "argName": "varLLL",
    "argType": "identifier",
    "changeable": 1,
    "className": "",
    "isClass": 0,
    "occurrenceNumber": 0,
    "isLeafNode": false,
    "positionInAST": [
        -1,
        -1.
        -1
    // ____17_ 1 ... // . .... 1 1
```

```
argvalue : null
} ,
    "grounded": false,
    "extended0perand1": {
        "argName": null,
        "argType": "number",
        "changeable": 0,
        "className": "",
        "isClass": 0,
        "occurrenceNumber": 1,
        "isLeafNode": true.
        "positionInAST": [
            -1,
            -1,
             1,
            -1
        "argValue": 10.0
    "extendedOperand2": {
        "argName": "x",
        "argType": "identifier",
        "changeable": 1,
        "className": "",
        "isClass": 0,
        "occurrenceNumber": 1,
        "isLeafNode": true,
        "positionInAST": [
            -1,
            -1,
             1,
        "argValue": null
    "extendedOperator": {
        "argName": "*",
        "argType": "other"
    },
    "extendedResult": [
        "argName": "varLLR",
        "argType": "identifier",
        "changeable": 1,
        "className": "",
        "isClass": 0,
        "occurrenceNumber": 0,
        "isLeafNode": false,
        "positionInAST": [
            -1,
            -1,
```

```
],
    "argValue": null
"grounded": false,
"extendedOperand1": {
    "argName": "varLLL",
    "argType": "identifier",
    "changeable": 1,
    "className": "",
    "isClass": 0,
    "occurrenceNumber": 1,
    "isLeafNode": false,
    "positionInAST": [
        -1,
        -1,
        --1
    "argValue": null
"extendedOperand2": {
    "argName": "varLLR",
    "argType": "identifier",
    "changeable": 1,
    "className": "",
    "isClass": 0,
    "occurrenceNumber": 1,
    "isLeafNode": false,
    "positionInAST": [
        -1,
        -1.
    "argValue": null
"extendedOperator": {
    "argName": "+",
    "argType": "other"
},
"extendedResult": {
    "argName": "varLL",
    "argType": "identifier",
    "changeable": 1,
    "className": "",
    "isClass": 0,
    "occurrenceNumber": 0,
    "isLeafNode": false,
    "positionInAST": [
        -1,
        -1
```

```
"argValue": null
},
    "grounded": false,
    "extendedOperand1": {
        "argName": null,
        "argType": "number",
        "changeable": 0,
        "className": "",
        "isClass": 0,
        "occurrenceNumber": 1,
        "isLeafNode": true,
        "positionInAST": [
            -1.
            1,
            --- ]
        "argValue": 1.0
    "extendedOperand2": {
        "argName": null,
        "argType": "number".
        "changeable": 0,
        "className": "",
        "isClass": 0,
        "occurrenceNumber": 1,
        "isLeafNode": true,
        "positionInAST": [
            -1,
            1,
        "argValue": 2.0
    "extendedOperator": {
        "argName": "+",
        "argType": "other"
    "extendedResult": {
        "argName": "varLR",
        "argType": "identifier",
        "changeable": 0,
        "className": "",
        "isClass": 0,
        "occurrenceNumber": 0,
        "isLeafNode": false,
        "positionInAST": [
            -1,
```

```
argvalue : null
},
    "grounded": false,
    "extendedOperand1": {
        "argName": "varLL",
        "argType": "identifier",
        "changeable": 1,
        "className": "",
        "isClass": 0,
        "occurrenceNumber": 1,
        "isLeafNode": false,
        "positionInAST": [
            -1,
            -1
        "argValue": null
    },
    "extendedOperand2": {
        "argName": "varLR",
        "argType": "identifier",
        "changeable": 0,
        "className": "",
        "isClass": 0,
        "occurrenceNumber": 1,
        "isLeafNode": false,
        "positionInAST": [
            -1,
        "argValue": null
    "extendedOperator": {
        "argName": "+",
        "argType": "other"
    "extendedResult": {
        "argName": "varL",
        "argType": "identifier",
        "changeable": 1,
        "className": "",
        "isClass": 0,
        "occurrenceNumber": 0,
        "isLeafNode": false,
        "positionInAST": [
        "argValue": null
},
```

```
grounded : laise,
        "extendedOperand1": {
            "argName": "varL",
            "argType": "identifier",
            "changeable": 1,
            "className": "",
            "isClass": 0,
            "occurrenceNumber": 1,
            "isLeafNode": false,
            "positionInAST": [
                --1
            "argValue": null
        },
        "extendedOperand2": {
            "argName": null,
            "argType": "number",
            "changeable": 0,
            "className": "",
            "isClass": 0,
            "occurrenceNumber": 1,
            "isLeafNode": false,
            "positionInAST": [
            "argValue": 0.0
        "extendedOperator": {
            "argName": "=",
            "argType": "other"
        "extendedResult": {
            "argName": "var",
            "argType": "identifier",
            "changeable": 1,
            "className": "",
            "isClass": 0,
            "occurrenceNumber": 0,
            "isLeafNode": false,
            "positionInAST": [],
            "argValue": null
"identifierSubstitutionTable": {},
"uniqueIdentifiers": [
    "varLLR",
    "varLL",
    \mathbf{x}''
    "varl",
    "var",
    "varLLLR",
```