牛顿迭代法说明文档

牛顿迭代法

1. **结构**

**Newton**为单个方程提供变量与解决方法

**NewtonWrappe**r对Newton进行包装以适应要求中多个方程的求解。并为后续有关该方程与功能的结构提供拓展性。

Node/Operator/Scalar 参考第一阶段说明文档

**2.具体封装、接口（Newton&NewtonWrapper）**

* Newton

private成员：

n：该方程次数

Timetag: 时间标签（用来避免在一次计算中重复计算已经计算过的节点）

eps: 用来处理float比较误差的极小值

ErrorSignal:处理错误信息

Nodes1, 变量的0-n次方节点(x^I，升幂)

Parameter 0-n次方的系数节点(ai，降幂)

Nodes2 0-n项节点（ai\*x^i，升幂）

Nodes3 0-n项求和（Si = a0\*x^0 +…+ ai\*x^i,升幂）

Node\* x :指向变量的指针

Node\* res： 指向建好的方程式节点的指针

Order\_of\_Derive 参考求导功能

public 成员

构造函数：Newton(int nn) 为参数n赋值

Construct\_Stage1-4分别按顺序建立Nodes1 Parameter Nodes2 Nodes3四个vector<Node\*>

Method:对图进行五次牛顿迭代法迭代

* NewtonWrapper

private成员

num:多项式个数

public 成员

NewtonWrapper(int) 构造函数，为num赋值

NewtonMathod() 按输入要求实例化一个Newton对象并调用Newton的各类接口

**3.说明**

在建图的过程中，我们顺原代码的第一阶段的编写思路，使用了ErrorSignal与Timetag，分别进行报错、防止重复计算。

Assign样例

3

yo V -0.3072

w V -2.8278

eu C 0.0000

4

a = sin yo

a = ASSIGN w yo

yo = eu \* a

k = ASSIGN w yo

3

EVAL k

0.0000

EVAL eu

0.0000

EVAL a

(-0.3072)

5

w C -4.8937

ny V -7.4334

dy P

fnp V 1.6042

ha P

5

o = PRINT ha

fnp = ASSERT fnp o

bou = PRINT dy

v =ny + fnp

v = ASSERT ny w

2

EVAL v

(-4.8937)

EVAL fnp 1 ha 3.78

(3.78)

4

q P

ov V 0.8483

rxo V 10.0000

gd P

5

k = ov \* rxo

g = ASSIGN ov k

x = gd + ov

r = PRINT x

lam = g + ov

3

EVAL r 1 gd 3.14

(3.9884)

EVAL lam

(9.3313)

EVAL r 1 gd 3.14

(11.623)

自动求导

Tensor计算图

Session

4

x V 1

y V 1

a C 3

b C 2

3

k1 = a \* x

k2 = b \* y

k = k1 + k2

12

APPLY NEW SESSION

SETCONST x 2

SETCONST y 3

EVAL k

SETANSWER x 4

APPLY NEW SESSION

SETCONST x 1

SETCONST y 0

EVAL k

SWTICH TO SESSION 1

EVAL k

SHOW SESSION

4

x V 1

y V 1

a C 3

b C 2

3

k1 = a \* x

k2 = b \* y

k = k1 + k2

11

APPLY NEW SESSION THROUGH FILE NAMED FILE1.txt

EVAL k

*//17*

SETANSWER x 2

APPLY NEW SESSION THROUGH FILE NAMED FILE2.txt

EVAL k

*//2*

SETCONST x 1

SETCONST y 0

EVAL k

*//3*

SWTICH TO SESSION 1

EVAL k

*//59*

SHOW SESSION

*//17 4*

4

x V 1

y V 1

a C 3

b C 2

5

dummy = ASSIGN x y

k1 = a \* x

k2 = b \* y

k = k1 + k2

res = BING k dummy

14

APPLY NEW SESSION THROUGH FILE NAMED FILE1.txt

EVAL res

*//*17

EVAL res

*//*20

APPLY NEW SESSION THROUGH FILE NAMED FILE2.txt

EVAL res

//2

EVAL k

//5

SETCONST x 1

SETCONST y 0

EVAL k

//3

SWTICH TO SESSION 1

EVAL res

//20

EVAL k

//20

SHOW SESSION

//4.4

FILE OUT OUT1.txt

//4 4

2

x V 0

a V 0

k C 3

4

temp = x \* k

in = ASSIGN a temp ;

temp = k \* in

res = temp + a

APPLY NEW SESSION THROUGH FILE NAMED FILE3.txt

FILE IN x 1

EVAL res

//9

EVAL in

//3

DELETE SESSION

SHOW SESSION

DELETE SESSION

梯度下降

手写识别