TBF程序设计说明书

一种深度学习pytorch 到地球系统模式的接口程序的程序设计说明书

# 一、引言

深度学习技术在近年来得到了广泛的应用，其在图像识别、自然语言处理、语音识别等领域取得了显著的成就。然而，深度学习技术在地球系统模式领域的应用相对较少。地球系统模式是一种复杂的模拟系统，用于模拟地球大气、海洋、陆地和冰冻圈等要素之间的相互作用。因此，将深度学习技术与地球系统模式相结合，可以为气候预测、环境保护等领域带来新的突破。

为了实现深度学习技术与地球系统模式的结合，我们设计了一种名为“深度学习pytorch 到地球系统模式的接口程序”的程序（torch bridge fortran ,简称TBF）。本说明书将介绍该程序的设计原理、功能特点、使用方法等内容。

# 设计原理

到目前为止，我们已经成功构建了一个计算边界层高度的参数化方法，接下来需要将这个模型加入到地球系统模式中，目前只有很少的工作是关于这方面的

我们建构建了一个torch和fortran代码的一个接口，fortran并不擅长人工智能类的编程，但是在大规模计算方面计算效率是很高的。我们的想法是，在现有框架（pytorch）下训练模型，再将训练好的模型架构和参数传递到用fortran 构建的大气海洋模式中。

在之前我们通过手动将一层神经网络放入过模式并取得成功，当时是将python传出参数等手动粘贴到fortran的代码当中，这也获得了成功，但是在可扩展性和便利性存在问题，这种手工的方法，每次训练完之后就需要手动将参数加进代码是很不方便的，如果多层成千上万的参数这是难以实现的。因此我们写了一个接口，python程序训练好模型后，会将模型参数存为txt,然后再由我们的TBF程序将参数传输进模式中，（具体的流程在流程图中展示，其中的计算过程在计算图中展示）这样大大方便了模型的改进，也使得巨大的参数数组能够轻易加入地球系统模式。

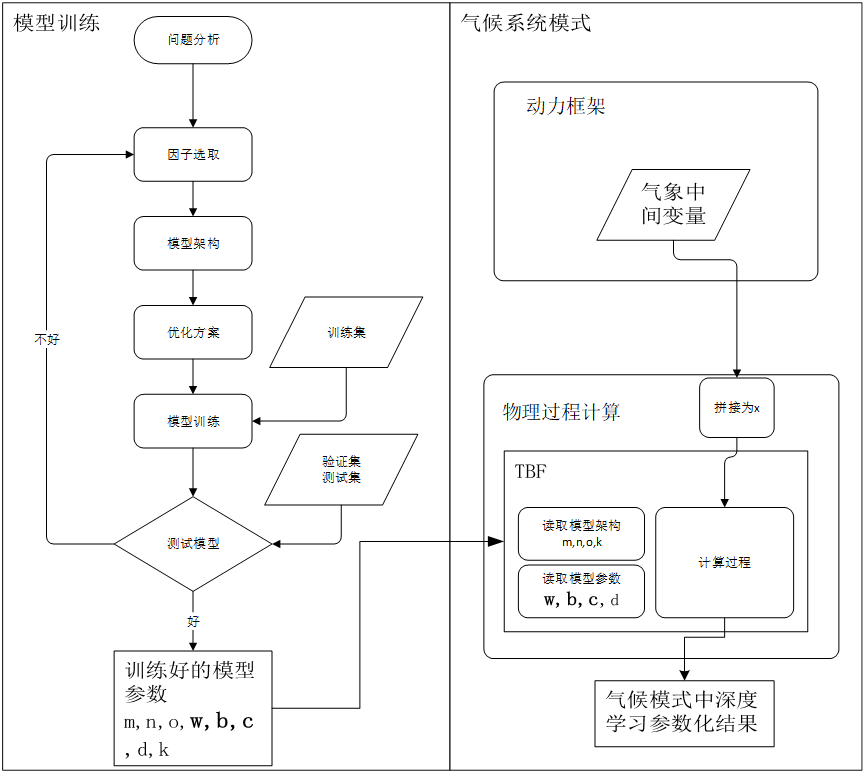


图 深度学习训练和加入模式流程图

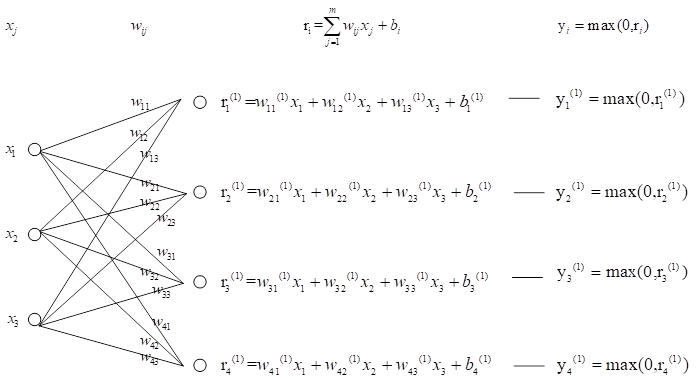
我们已经选择使用多层前馈神经网络，因此我们主要的网络架构已经知道了，我们在fortran中重构计算方法，训练好的模型参数输入进来就可以计算输出

我们构建的接口，只适用于深度前馈网络（或者全连接网络，DNN）多因子多层回归，而更多的功能会在需要时添加。

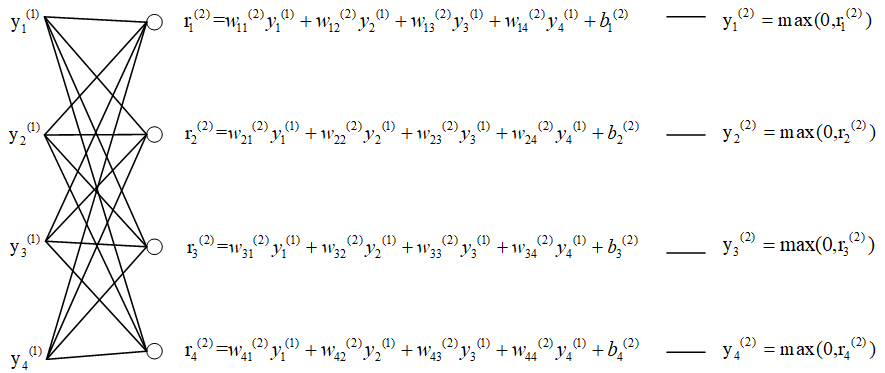
深度学习模型架构,假设有m 个因子，有o层，每层都有n 个计算节点。

我们把输入x设成有m 个元素的向量x(m) ，索引 j = 1:m,把W设计成一个三维数组W(n,m,o),w(i,j,k),k表示第k层，k =1: o;i 表示这层第i个节点，i = 1: n;j表示与第j个因子相乘的权重；把b 设成一个二维数组b(n,o),b(i,k),索引i表示第i个节点上的偏差，k表示第k层上的偏差；把c设成一个一维向量c(n),索引i= 1,n表示第i个节点上的权重；把d设成一个标量。

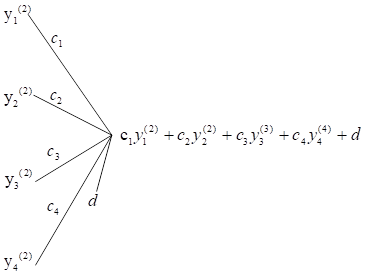
下面我们假设有三个因子，设置每层的节点数为4，两层前馈神经网络（更多层的情况结合公式-推导），具体的推导见下图



输入层计算图



中间层计算图



输出层计算图

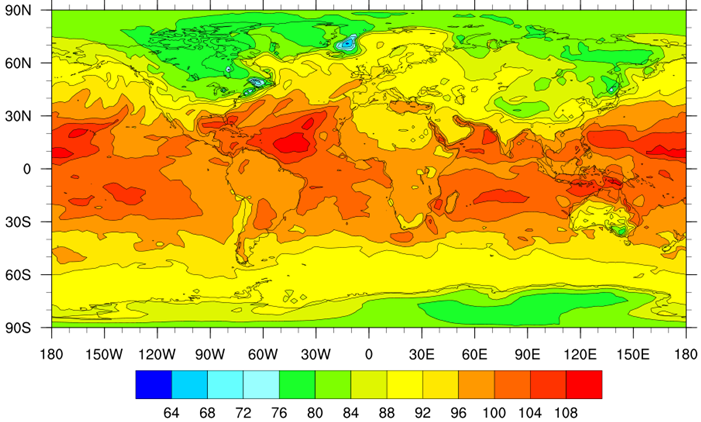


多层示意图 草稿（全）

更多的TBF有关的内容，可以看TBF 的手册。具体的代码在github中https://github.com/2469488573/torch\_to\_fortran-

**在地球系统模式中使用TBF**

为了加入到地球系统模式中，我们写了TBF的module，使用者可以将module引用，然后只需要指定深度学习训练目录和几个计算因子，然后调用计算函数就可以计算。



**图** 深度前馈神经网络边界层高度参数化方法加入模式的输出结果

深度学习pytorch 到地球系统模式的接口程序的设计原理主要包括以下几个方面：

# 三、功能特点

深度学习pytorch 到地球系统模式的接口程序具有以下功能特点：

1. 灵活性：接口程序可以适配不同的地球系统模式和深度学习模型，具有较强的通用性。
2. 高效性：接口程序经过算法优化，能够在大规模数据和复杂模型的情况下保持较高的计算效率。
3. 可扩展性：接口程序具有较强的可扩展性，可以根据需要进行功能扩展和性能优化。

# 四、使用方法

深度学习pytorch 到地球系统模式的接口程序的使用方法主要包括以下几个步骤：

## **1.使用Python程序训练深度学习模型**

这个程序是一个TBF的实例程序的torch部分，

### 第一步：需要输入训练数据，验证数据，测试数据，

1.计算一个y所需要的x1，x2...xm,这个是m = feature\_count .

### 第二步：网络架构设计：设置有几层，每层有几个节点，哪类激活函数

1.每层有几个节点在point\_count,feature,n = point\_count 设置；

2.有几层这个在nn.sequential{}中设置，然后在后面的o = {激活函数的个数}

3.在nn.sequential 中激活函数和线性层交替排列，现代深度学习一般relu为激活函数 也可以使sigmod、tanh等，这个在torch最好设为一样的，在KBF目前只能设置成一样的。

### 第三步：设置优化方法，主要是对优化方法，npoch,batch\_size,等进行修改，以达到最好的优化效果，得到最优参数W,b,c,d

### 第四步：查看训练的结果，两张图，方差，均方根误差等决定需不需要再次重复上面步骤。

### 第五步：在训练效果很好，得到正确参数的情况下，传输模型参数先输出为txt文件，

然后用fortran读取到KBF中。

1.检查m,n,o,function\_kind,shuchucanshu.txt

2.检查w\_input.txt w\_dense.txt w\_output.txt

3.检查b\_input.txt b\_dense.txt b\_output.txt

### 第六步：将\*.txt 传输到torch\_bridge\_fortran 文件夹中

之后的操作留给fortran程序来解决。

## **2. 使用TBF接口程序将深度学习模型放进地球系统模式进行计算**

!torch\_bridge\_fortran 被写成了module，方便在其他程序中调用

!其中使用者可以把它当成黑箱，深度学习训练好的模型被封装在其中

!需要输入的是一个二维数组，x\_array(cesm\_m,ncol),cesm\_m 是因子个数

!ncol和cesm的ncol是一致的。输出的y\_cesm就是深度模型的计算量

!x1, i --| \_\_\_\_\_\_

!x2, i --| | |

!x3, i --|----------> | box | -------> y\_cesm(i)

!x4, i --| |\_\_\_\_\_|

!x5, i --|

!这是cesm中调用TBF的一个实例程序，在bridge程序中，计算被进行

!参数传递的过程在bridge进行，

!参数优化的过程则是在python文件夹下进行

!计算过程则主要写在calculation.f90中，这一部分不建议修改，需要对深度模型很了解

!文件读取在file\_io.f90中进行

!----------------------------------------------------------------

!在fortran模型代码中，我们主要需要做下面几件事

### !1.use module

use bridge , only: tbf

!-----------------------------------------------------------------

### !2.申明变量和变量的维度

implicit none

integer :: m = 4

real,allocatable :: x\_array(:,:)

real :: y\_cesm(5)

character(len = 100) :: dirname ="/data/chengxl/&

pblh\_deeplearning/torch\_bridge\_fortran/python/"

!-----------------------------------------------------------------

### !3.明确自变量（可以通过其他函数传递）!如果多个变量向量则需要进行数组拼接

allocate(x\_array(m,5))

x\_array(:,1) = (/264.32004,0.3210011,14510.625,52310.562/)

x\_array(:,2) = (/264.31717,0.32086015,14449.125,52227.875/)

x\_array(:,3) = (/264.31717,0.32067218,14449.125,52186.5/)

x\_array(:,4) = (/264.31573,0.3205077,14449.125,52062.375/)

x\_array(:,5) = (/264.31573,0.3203667,14387.5,51979.688/)

!------------------------------------------------------------------

### !4.调用子程序tbf( 数组长度，因子个数，自变量数组, 预报量数组)

!增加功能，传进去一个路径参数使得这个tbf能够被多次调用

call tbf(dirname,5,m,x\_array,y\_cesm)

!------------------------------------------------------------------

### !5.检查和传递计算结果

print\*,y\_cesm

!-------------------------------------------------------------------

print\*,"测试"

这样深度学习模型已经被加入Fortran编写的模式中计算了。

# 五、总结

深度学习pytorch 到地球系统模式的接口程序的设计说明书介绍了该程序的设计原理、功能特点、使用方法等内容。该接口程序的设计旨在实现深度学习技术与地球系统模式的有效结合，为地球科学领域的研究和应用提供新的思路和方法。希望该接口程序能够为相关领域的研究人员和工程师提供便利，推动深度学习技术在地球系统模式领域的应用和发展。