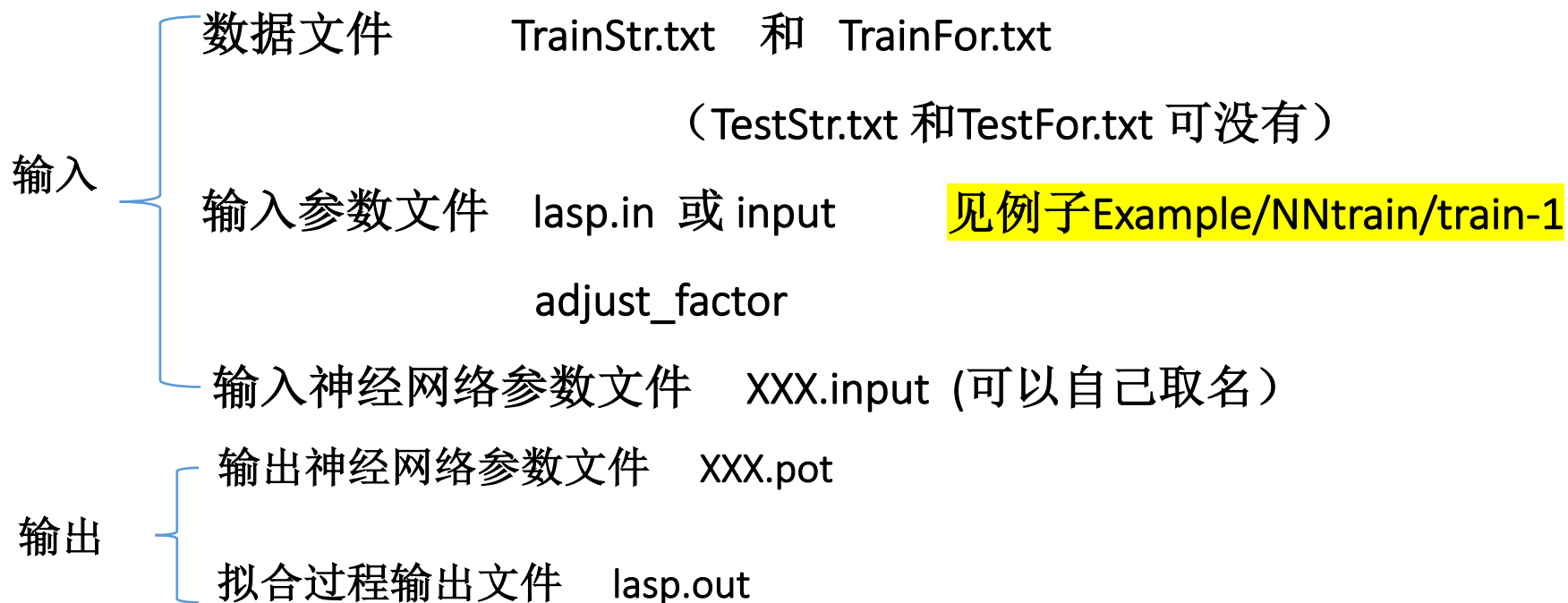


## LASP 使用入门11 – 构建神经网络势函数 Available from LASP 2.0

**目标：** 基于第一性原理计算的数据集，通过训练（参数优化）得到神经网络势函数



## LASP 使用入门11 – 构建神经网络势函数 LASP 2.0

**数据文件**: TrainStr.txt 结构 和 TrainFor.txt 力/应力

1. 通过SSW或MD等方法，得到allstr.arc和allfor.arc（见使用入门第五讲：拟合全局神经网络势函数-准备数据）。

如果有很多不同的allstr.arc, allfor.arc, 可以分别粘到一起，形成最后的allstr.arc和allfor.arc，需格外注意结构的顺序两个文件必须对应

2. 利用python脚本 shiftformat.py 将allstr.arc和allfor.arc 互相转化为TrainStr.txt和TrainFor.txt  
(见Example/NNtrain/data-treat)

shiftformat.py 需要 python 2.7（及以后版本）支持

## LASP 使用入门11 – 构建神经网络势函数

- TrainStr.txt

### Structure & energy

```
[root@storage3 train-1]# head -100 TrainStr.txt
Start one structure
Energy is -90.311342 eV
total number of element is 21
element in structure:
symbol      H      0
No.         1      8
number     14      7
weight     1.000   1.000
lat        3.35380000   0.00000000   0.00000000
lat       -0.07946898   6.22929312   0.00000000
lat        0.02404341   0.15952898   3.76974942
ele 1      0.14484300   1.44770700   3.03983500   0.00000000
ele 1      1.85554600   6.14001300   3.26371600   0.00000000
ele 1      3.11433900   5.02752500   3.23655200   0.00000000
ele 1      1.78860000   4.76133200   2.51173200   0.00000000
ele 1      0.86479900   1.99063900   0.72217300   0.00000000
```

- TrainFor.txt

### Stress & Force

```
[root@storage3 train-1]# head -100 TrainFor.txt
Start one structure
stress     -0.42497900  -0.02932100  -0.13619800  -0.47462300   0.04716500  -0.66523600
force 1    -0.05439800   0.24378700   0.02063000
force 1     0.28527200  -0.12513800  -0.25176500
force 1    -0.28888800  -0.00618900   0.16100700
force 1    -0.46466100   0.08383700  -0.17050500
force 1     0.07224600  -0.39832200  -0.01284800
force 1    -0.11602100  -0.79190800   0.00023400
force 1     0.36663400   0.17577500  -0.58945900
force 1     0.55632000   0.45096100   0.42107600
force 1    -0.00741000  -0.36140500   0.11907300
force 1    -0.15219600   0.04598200   0.49314600
force 1    -0.18696600   0.12952600   0.55094900
```

## LASP 使用入门11 – 构建神经网络势函数

输入参数文件 `lasp.in` 或 `input`

```
[root@storage3 train-1]# cat lasp.in
#
explore_type train
Ntrain 100
# Input of training
#
Jobname      H2O
Ntest        0
NNepochs     200
readScale    0 # decide read Gmin/Gmax
analyzesym   0
#
# Specify network information
#
%block netinfo
H  H2O.input
O  H2O.input
%endblock netinfo
```

关键参数如下 (`allkeys.log` 列出所有参数)

**explore\_type train**      关键词 **train**

**Ntrain 100**              训练**100**个数据结构

**NNepochs 200**          训练**200**代

实际体系中，结构数据一般都是几万  
训练需要**1万代以上**

**%block netinfo**  
每个元素的**NN**参数文件

## LASP 使用入门11 – 构建神经网络势函数

输入参数文件 `adjust_factor`

```
[root@storage3 train-1]# cat adjust_factor
calfact 200 1000.0 10 0
outfact 1.0 0.0 0.0
train_RMS 1
test_RMS 0
Time_step 1
BFGS_loop 1
```

这个文件一般不需要修改，第一行最重要

**200 1000 10 0**

前三个数**200/1000/10**控制了训练的总**cost function**中  
**energy, force, 和stress**的构成相对权重

# LASP 使用入门11 – 构建神经网络势函数

输入神经网络参数文件 XXX.input

```
[root@storage3 train-1]# head -100 H2O.input  
shiftElement 0 2 1 8  
  
Element 0  
Neural Network Architecture  
# Note : typically, fully layer is putting in first/final layer  
%block 0_netinfo  
# fully layer -> full, size, active_type, float/fixed, newrun/reuse-filename  
# input 106 2 float newrun  
  full 50 2 float newrun  
  full 50 2 float newrun  
  full 1 0 float newrun  
%endblock 0_netinfo
```

元素解释  
将0，解释为包含2种元素，分别为1（H），8（O）  
见下页

神经网络的架构 106-50-50-1  
50-50为中间的隐藏层

首次train，用newrun  
如果restart，需要改为相应神经网络参数的XXX.input（如：复制以前的pot）

## LASP 使用入门11 – 构建神经网络势函数

contin: 输入神经网络参数文件 XXX.input      PowerType Structure Descriptors(PTSD)

包含了所有的PTSD的基本参数，分为 S1, S2, S3, S4, S5, S6 (公式详见LASP 相关论文)

```
S1 : sum(rij**n * fc(rij))
    : radical distribution, large n means considering more long-distance neighb
%block 0_S1
#-----#
# cutoff   neighb_atom   n           Gmin           Gmax           #
#-----#
1.20       1           8           0.0000000000000000E+00   0.183035038932026E-01 # 1
1.40       1           2           0.0000000000000000E+00   0.103357295876206E+00 # 2
1.80       1           4           0.0000000000000000E+00   0.426574772070577E+00 # 3
1.80       8           8           0.0000000000000000E+00   0.184514583370168E+00 # 4
1.90       8           2           0.0000000000000000E+00   0.955976164742935E-01 # 5
1.80       8           4           0.0000000000000000E+00   0.836717650519582E-01 # 6
2.00       0           -3          0.0000000000000000E+00   0.147432709026505E+01 # 7
3.00       0           -3          0.0000000000000000E+00   0.199326610771814E+01 # 8
4.00       0           -3          0.0000000000000000E+00   0.228723664653087E+01 # 9
```

1: H; 8: O 元素

NN输入参数

0: 这里代表所有元素，由 shiftElement 关键词解释（上页）

## LASP 使用入门11 – 构建神经网络势函数

输出神经网络参数文件 XXX.pot ,即是最后LASP可以使用的NN参数文件  
文件头包含了元素, 时间, 数据集大小, 拟合精度的基本信息

```
Neural Network Potential For LASP
Reference   : www.lasphub.com
Name       :
Species    : Mn H O
Version    : 20190224063759
Dataset size : 14239
```

Fitting Accuracy

	Energy/(meV/atom)	Force/(eV/A)	Stress/(GPa)
RMS	2.903	0.129	1.841
MAX	34.982	4.455	17.079
scale_RMS	2.903	0.129	1.841
scale_MAX	34.982	4.455	17.079



# LASP 使用入门11 – 构建神经网络势函数

训练过程输出文件 lasp.out

```
-----+-----+
+   train neural network potential   +
+-----+-----+

NN Started at 22:45:35 23rd February 2019

Time consuming until read train data is 0.0472 s

Time consuming for build network is 0.0476 s

Number of parameters in system is 15902

H : layer 1 is inpu with 1 channel, each 106 nodes, total 106 nodes.
H : layer 2 is full with 1 channel, each 50 nodes, total 50 nodes.
H : layer 3 is full with 1 channel, each 50 nodes, total 50 nodes.
H : layer 4 is full with 1 channel, each 1 nodes, total 1 nodes.
O : layer 1 is inpu with 1 channel, each 106 nodes, total 106 nodes.
O : layer 2 is full with 1 channel, each 50 nodes, total 50 nodes.
O : layer 3 is full with 1 channel, each 50 nodes, total 50 nodes.
O : layer 4 is full with 1 channel, each 1 nodes, total 1 nodes.

YWarning : 0 S1 No. 4 Gmax is too small in dataset.
YWarning : 0 S1 No. 6 Gmax is too small in dataset.

Time consuming for calculating symf is 10.7626 s

-----+-----+

step 0 C = 3414177.565629 BFGS-step = 0.000000
```

NN总参数大小

参数越多，拟合精度越高  
(也容易过拟合)

表述数据集不够大，有神经网络PTSD没有值

训练目标函数 (cost function) 的数值

越小越好

## LASP 使用入门11 – 构建神经网络势函数

### 一些注意事项

1. 神经网络势函数建立在大数据基础上，单个数据不宜原子数太多（如 $<200$  原子，否则拟合不了）

小数据（MD数据）等能拟合精度高，不代表具有迁移性

2. SSW全局数据拟合，一般需要选择适当的范围，比如 energy 不能是正的，force 不大于  $100 \text{ eV/\text{Ang}}$

数据集DFT计算的精度需要高并且自洽，目前LASP使用的NN pot，基本由VASP在PBE（或PBE+U）在  $450 \text{ eV cutoff}$ ，自动K-point  $25 \text{ Ang}^{-1}$  产生

3. NN拟合一般用大量CPU/core MPI并行

LASP使用的NN pot 通过迭代自动训练得到，一般使用  $1000 \text{ cores}$ ，一周以上时间得到

## LASP 使用入门11 – 构建神经网络势函数

### 一些注意事项

#### 4. PTSD 结构描述函数

- 均匀**: 每种PTSD, S1-S6, 都写; Cutoff短长都写; 角度都有
- 有限次方**: 幂次方, 一般在 -3, 2, 4, 8, 16, 32中选
- 避免长S5和S6**: 三体S5和四体的S6, 计算代价高, 不建议采用长程的截断半径 (Cutoff, 如 7 Ang)
- 4-5层神经网络**: 几万参数如 200-50-50-1; 可加大网络 200-80-50-50-1