MTPR potentials description

MTPR (moment tensor potentials radial) are extension of MTP potentials to the case of several components.

MTPR usage scenarios

Training of MTPR potentials

make command: make `learn_mtpr`

This procedure has as its aim obtaining of the MTPR potential able to give accurate EFS predictions for configurations which lie within the convex hull, formed by configurations from training set in configurational space of atom coordinates and types.

Takes 5-7 arguments: mpiexec -n X `./MTPR_train_exe` \$1 \$2 \$3 \$4 \$5 \$6 \$7
X - number of cores
\$1 - potential name
\$2 - training set name
\$3 - energy coefficient for training
\$4 - forces coefficient for training
\$5 - stresses coefficient for training
\$6 - validation set name (optional, default = none)
\$7 - number of BFGS iterations while training (optional, default = 2000)

For serial mode, launch as `./MTPR_train_exe` \$1 \$2 \$3 \$4 \$5 \$6 \$7

Output files:
`Trained.mtp_` - the resulting trained potential
curr.mtp - potential at the current iteration (its name and existence is
specified at the constructor of MTPR trainer).

Errors mode:
If (\$3==0)&&(\$4==0)&&(\$5==0), which means all weights are equal to 0, then the errors on the training (or also on validation) set are calculated and no output files are produced.

Active selection of configurations

make command: make `select_mtpr`

In this scenario the so-called multiple selection is implemented and the amount of selected configurations is limited to 100.

Takes 3 arguments: `./MTPR_select_exe` $1 $2 $3
$1 - potential name
$2 - set to select from
$3 - set to initialize the MaxVol

Output files:
Select.cfgs - all selected configurations
`TS_selected` - selected configurations, excluding ones present in the $3
If `($3==TS_selected)`, then this file will be extended with new selected configurations
valid.cfgs - $2 excluding `TS_selected`
state.mvs - the state of the MaxVol object at the end of selection, can be used for MaxVol initializing by other MTP utilites

Relaxation with MTPR potential

make command: make `relax_mtpr`

The following working folders are created: unrelaxed, relaxed, preselected, 4relax. These folders are needed for a proper parallelization. Each thread will have its own file in each of these folders.

Takes 0 arguments: mpiexec -n X `./MTPR_relax_exe`
X - number of cores
For serial mode, launch as `./MTPR_relax_exe`

Input files:
pot.mtp - potential to relax with
to-relax.cfg - configurations to relax
train.cfg - set to initialize the MaxVol

Output files:
relaxed.cfg - successfully relaxed configurations
preselected.cfg - configurations with grade exceeding the threshold
selected.cfg - configurations, selected by the MaxVol from preselected.cfg
unrelaxed.cfg - configurations failed to relax

Examples

In the folder `mlip_source/dev_doc/examples/` you will find three folders corresponding to different MTPR usage scenarios: `MTPR_relax`, `MTPR_select`, `MTPR_train.`

To launch each example you need to compile appropriate binary file, put it in the folder and execute a shell script from this folder. Using the provided input files should give the same output as in `sample_output.txt` file.

Running main relax+select+train.sh or md+select+train.sh script on cluster

1. Copy public MLIP to cluster in home directory.
2. Go to `mlip/make folder`, enter 'make mlp'. After that 'mlp' binary will be generated into mlip/make folder. This file is necessary to convert .cfgs file into POSCARs and OUTPUTs into .cfgs file.
3. Go to `mlip-dev/dev_make` folder in developer MLIP. Generate three binaries described above. Moreover, generate `Dump_to_CFGS` binary by making command:
make `dump_to_cfgs.`
This binary file is used in md+select+train.sh script to convert LAMMPS dump file to MLIP .cfgs file. After that, copy all binaries mentioned above to directory with relax+select+train.sh and md+select+train.sh scripts.
4. Run relax+select+train.sh (or md+select+train.sh) script as follows (the script has three input parameters):

```
./relax+select+train.sh $1 $2 $3
```

$1 - remote machine name,
$2 - path to mlp binary from cluster home directory (without / at the begin and at the end!),
$3 - number of nodes that will be used to calcute VASP OUTPUTs simultaneously

Examples of run:

```
./relax+select+train.sh username@10.30.16.62 mlip/make 7
./md+select+train.sh User.Name@pardus.skoltech.ru mlip/make 7
```