Aenet changes

Mário R. G. Marques, Miguel A. L. Marques

April 24, 2019

1 Changes - version 1.0.0

We made some changes in the aenet code.

We implemented 3 activation functions: Relu, sofplus, and a gaussian function. Their keywords are relu, splus, and gauss, respectively.

We added the calculation of the stress tensor to prediction.x.

During our calculations we found that it might be convenient to include in our objectives a dimensionless weight factor w_i that is given in terms of $\bar{F}_i$ the average norm of forces acting on atoms in the i^{th} structure. It reads, when $\bar{F}_i$ is measured in eV/Å,

$$w_i = \frac{0.2}{0.2 + \bar{F}_i^2} \, . \tag{1}$$

This can be done by adding the variable **structure_force_weight** in the input for the generate.x. The RMSE and MAE will then be computed according to the sum of these weights. If the variable is not set, the errors will use the total number of structures.

The cost function in the code only took into account the errors in the energy. We implemented the necessary equations to extended it, in order to also include forces and stress errors.

$$\begin{aligned} \epsilon \; = \; \sum_\sigma \Bigg[& \alpha \left(\sum_i^N E_i - E^{\text{ref}} \right)^2 \; + \; \frac{\gamma}{6} \; \sum_k^6 \left(S_k - S_k^{\text{ref}} \right)^2 \\ + \; & \frac{\beta}{3N} \; \sum_\gamma^N \sum_\alpha^3 \left(F_{j\alpha} - F_{j\alpha}^{\text{ref}} \right)^2 \Bigg]_\sigma \end{aligned} \tag{2}$$

The variable WOBJECTIVES can be set in the input for the train.x to change the values of α, β, and γ. Example:

```
WOBJECTIVES alpha=0.4 beta=1 gamma=0.4
```

The variable MODE defines the terms to be included in the cost function:

- E stands for energy training and is the default value for MODE.

- F stands for forces training
- S stands for forces training
- ESF stands for energy, forces and stress training

Example:

```
MODE ESF
```

The number of iterations can be set with

```
ITERATIONS 20
ITERATIONS_FORCES 0
ITERATIONS_STRESS 0
ITERATIONS_LOOP 1
```

If the energy training is selected, the code will perform ITERATIONS iterations for the energy training, ITERATIONS_FORCES iterations for the forces and ITERATIONS_STRESS iterations of the stress training. Which can be repeated ITERATIONS_LOOP times. We change this in the 2 version.

In order to have a good force field we have to sample different regions of the potential energy surface. However, some regions (or a specific type of structure) might be more important than others. It might be necessary to increase the weight of the reference structure in order to compute precise formation energies.

The block STRUCTURE_WEIGHTS can be set in the input for the train.x to increase the importance of these structures. Each row of the block takes a pattern and a value for the weight. These values are used during training, they are not included in the calculation of the errors.

```
{STRUCTURE_WEIGHTS
ref       100
minima    100
2D        100
distorted 1
md        1
}
```

During testing of the neural networks, we found out something that is actually quite trivial. If structures with atoms that are too close to each other are not included in the training set, the force fields might behave unphysically. This can be solve by adding such structures to the training set. However, these structures will have high energies, which might hinder the training. Another option is to include a repulsive potential. We choose a potential of the form

$$f(x) = \epsilon \left(\frac{r_0}{r}\right)^{12} \tag{3}$$

The parameters can be set in the input for the generate.x and predict.x:

```
REPULSIVE_EPSILON 1.00
REPULSIVE_RZERO   2.00
```

To include them in the training one should use the variable REPULSIVE in the input for the train.x

```
REPULSIVE
```

The variable USE_ASCII can be use to write the neural network files (.ann or .nn) in normal readable files.

```
USE_ASCII
```

Additionally, we added the possibility to use ascii v_sim files and mlcif. In ascii v_sim files the forces and stresses should be commented with # while in the mlcif, the first line should have the number of atoms, energy and the stress tensor in Voigt notation, followed by the lattice vectors. Afterwards, there should be the positions of the atoms, one atom per line, and with the chemical symbol at the beginning. The forces must follow the positions, 1 force per line. AS an example:

```
3 -9.834014 -0.003528 -0.001375 -0.004386 -0.002385 -0.001008 -0.001923
2.937229 -0.020461 -0.092911
-1.470175 6.473182 -0.022128
0.064054 -0.662882 2.863197
Au -0.368696860888 1.32804170629 2.4147676056
Au -0.412439642459 6.07859748015 0.460247682961
Au 1.07804637134 3.69308916898 1.39105202458
1.369e-05 -0.00012989 -1.869e-05
-3.18e-06 0.00017584 4.258e-05
-1.051e-05 -4.594e-05 -2.389e-05
```

Finally, we define a new executable: analyse.x, which works like the predict.x, however it prints forces, energies ad stresses from the reference as well as the neural network calculated quantities.

2 Changes - version 2.0.3

We did not implement everything in the second version. The repulsive potential was not included yet and we changed the way MODE works.

- E stands for energy training and is the default value for MODE
- F stands for forces training
- S stands for forces training
- EF stands for energy and forces training

- ES stands for energy and stress training
- SF stands for forces and stress training
- ESF stands for energy, forces and stress training

There is only one variable for the iterations.

In this version the additional training modes should only be used with the BFGS method.