Computation of Atomic Descriptors in HALMD Integration of DeepMD-v2 Potentials

Sandip Kumar Sah

Computation of Atomic Descriptors

After the normalized environment matrices $\mathbf{R}^{(i)}$ are constructed, they are passed through the **embedding (filter) networks** to generate atomic-level features $\mathbf{G}$. Each central atom i is associated with multiple neighbor types (e.g., A, B, ...), and each pair (t_c, t_n) of *center type* and *neighbor type* has a dedicated embedding MLP with shared parameters across atoms of the same type.

Feedforward Embedding Network with Residual Rule

For a given pair (t_c, t_n), the filter network consists of a stack of fully connected layers:

$$\mathbf{h}^{(0)} = \mathbf{x}, \quad \mathbf{h}^{(l+1)} = \tanh\big(\mathbf{W}^{(l)}\mathbf{h}^{(l)} + \mathbf{b}^{(l)}\big),$$

followed by an optional residual addition:

$$\mathbf{h}^{(l+1)} \mathrel{+}= \begin{cases} \mathbf{h}^{(l)}, & \text{if } \dim(\mathbf{h}^{(l+1)}) = \dim(\mathbf{h}^{(l)}), \\ [\mathbf{h}^{(l)}, \mathbf{h}^{(l)}], & \text{if } \dim(\mathbf{h}^{(l+1)}) = 2 \times \dim(\mathbf{h}^{(l)}). \end{cases}$$

This rule, called the **residual rule**, is an inherent property of DeepMD v2 and is automatically applied whenever the output and input dimensions satisfy one of the above conditions. It stabilizes training and ensures better gradient flow, even though it is *not explicitly specified* in the model configuration (`input.json`). Hence, the same residual behavior must be reproduced exactly during inference in HALMD.

Difference from a classical ResNet: Unlike conventional residual networks (ResNets), which define explicit residual blocks ($\mathbf{y} = f(\mathbf{x}) + \mathbf{x}$), DeepMD applies this rule *per layer* based solely on dimensional matching. It has no separate projection layers or normalization operations. The mechanism is thus simpler, shape-driven, and tailored for molecular descriptors.

Shape of Variables

For each central atom i:

- Normalized environment matrix: $\mathbf{R}^{(i)} \in \mathbb{R}^{N_{\text{neigh}} \times 4}$ (4 columns: inverse distance and scaled components).
- Embedding network output: $\mathbf{G}^{(i)} \in \mathbb{R}^{N_{\text{neigh}} \times M_1}$, where M_1 is the output width of the final embedding layer.
- Axis selection: $\mathbf{G}_{\text{lt}}^{(i)} = \mathbf{G}^{(i)}[:, : M_2]$ selects the first M_2 columns (`axis_neuron`).

Descriptor Evaluation in HALMD

HALMD computes the atomic descriptor directly using explicit matrix multiplications rather than tensor contractions. The mathematical expression implemented in HALMD is:

$$\mathbf{D}^{(i)} = \frac{1}{N_{\text{neigh}}^2} \mathbf{G}^\top \left(\mathbf{R}\mathbf{R}^\top\right) \mathbf{G}_{\text{lt}}.$$

The computation proceeds as:

$$\begin{aligned}
\mathbf{R}\mathbf{R}^\top &\in \mathbb{R}^{N_{\text{neigh}} \times N_{\text{neigh}}}, \\
\mathbf{tmp} &= (\mathbf{R}\mathbf{R}^\top)\mathbf{G}_{\text{lt}}, \\
\mathbf{D}^{(i)} &= \frac{1}{N_{\text{neigh}}^2}\mathbf{G}^\top\mathbf{tmp}.
\end{aligned}$$

Finally, the descriptor $\mathcal{D}_i$ is obtained by flattening $\mathbf{D}^{(i)}$ into a one-dimensional vector:

$$\mathcal{D}_i = \text{vec}(\mathbf{D}^{(i)}) \in \mathbb{R}^{M_1 M_2}.$$

Implementation Notes:

- Although this approach requires more intermediate matrix operations than the original DeepMD tensor-based formulation, it is designed for readability and compatibility with HALMD's internal data structures.
- The explicit computation of $(\mathbf{R}\mathbf{R}^\top)$ simplifies derivative calculations required for computing atomic forces.
- The descriptor formation remains mathematically consistent with DeepMD v2, ensuring identical force and energy predictions when the same parameters are used.

Shape of the Final Descriptor

For each atom i:

$$\mathcal{D}_i \in \mathbb{R}^{M_1 \times M_2} \quad \text{(matrix form)}, \qquad \text{or equivalently } \mathcal{D}_i \in \mathbb{R}^{M_1 M_2} \text{ (flattened vector)}.$$

Typical parameter values (from trained alloy models) are:

$$M_1 = 100, \quad M_2 = 16\text{–}32, \quad N_{\text{neigh}} \approx 100.$$

These descriptors are then passed to the fitting network to predict atomic energy contributions for multi-species alloy systems.