



۲nd Workshop on پژوهیش کارگاه

پایه‌گری ماشینی در فیزیک: Machine Learning in physics:

کاربردها در ماده چال Applications in Condensed Matter Physics

Machine Learning in Condensed Matter: Generating Potential Surface Energy with ANN

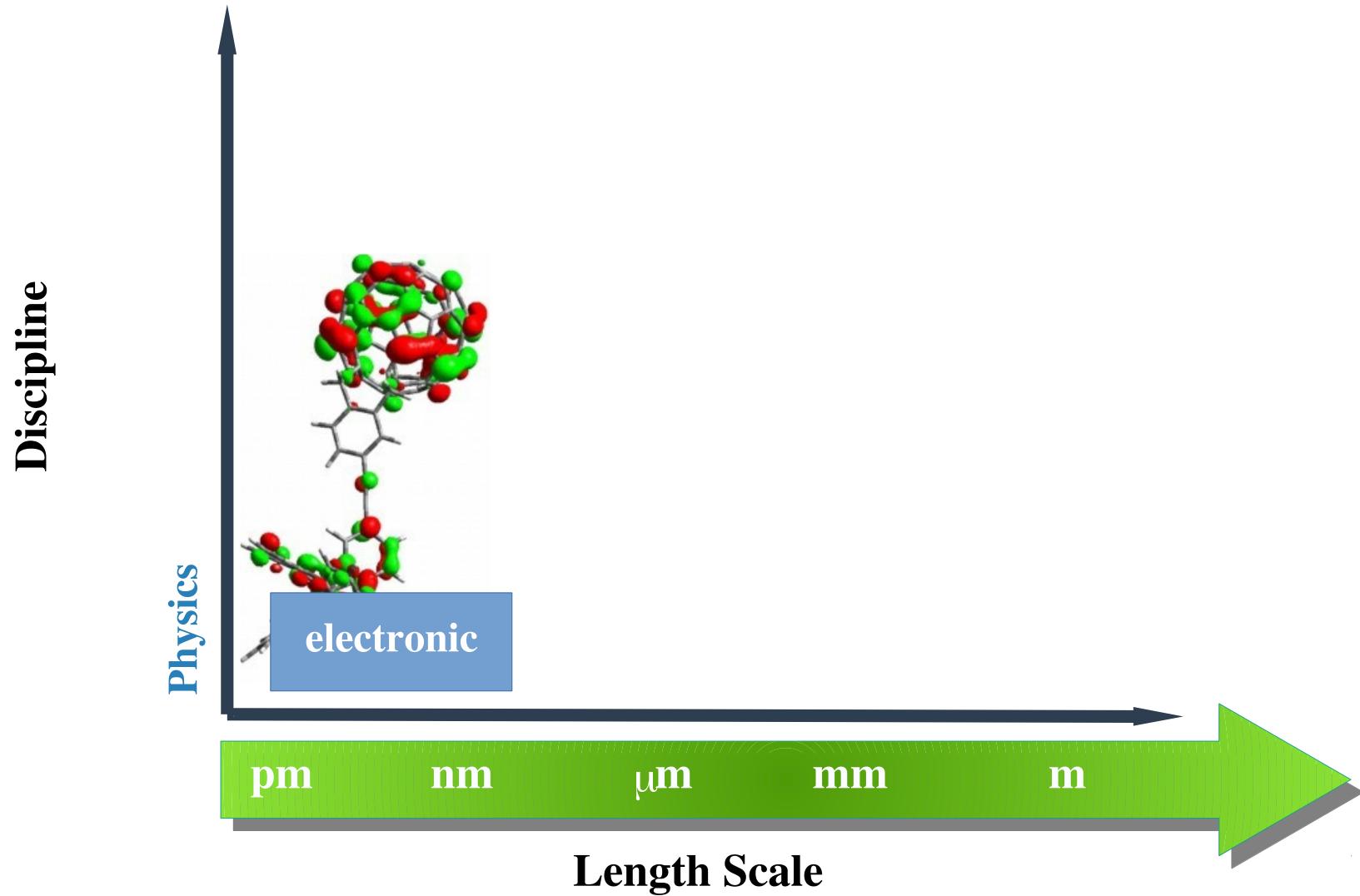
Robabe Rasoukhani



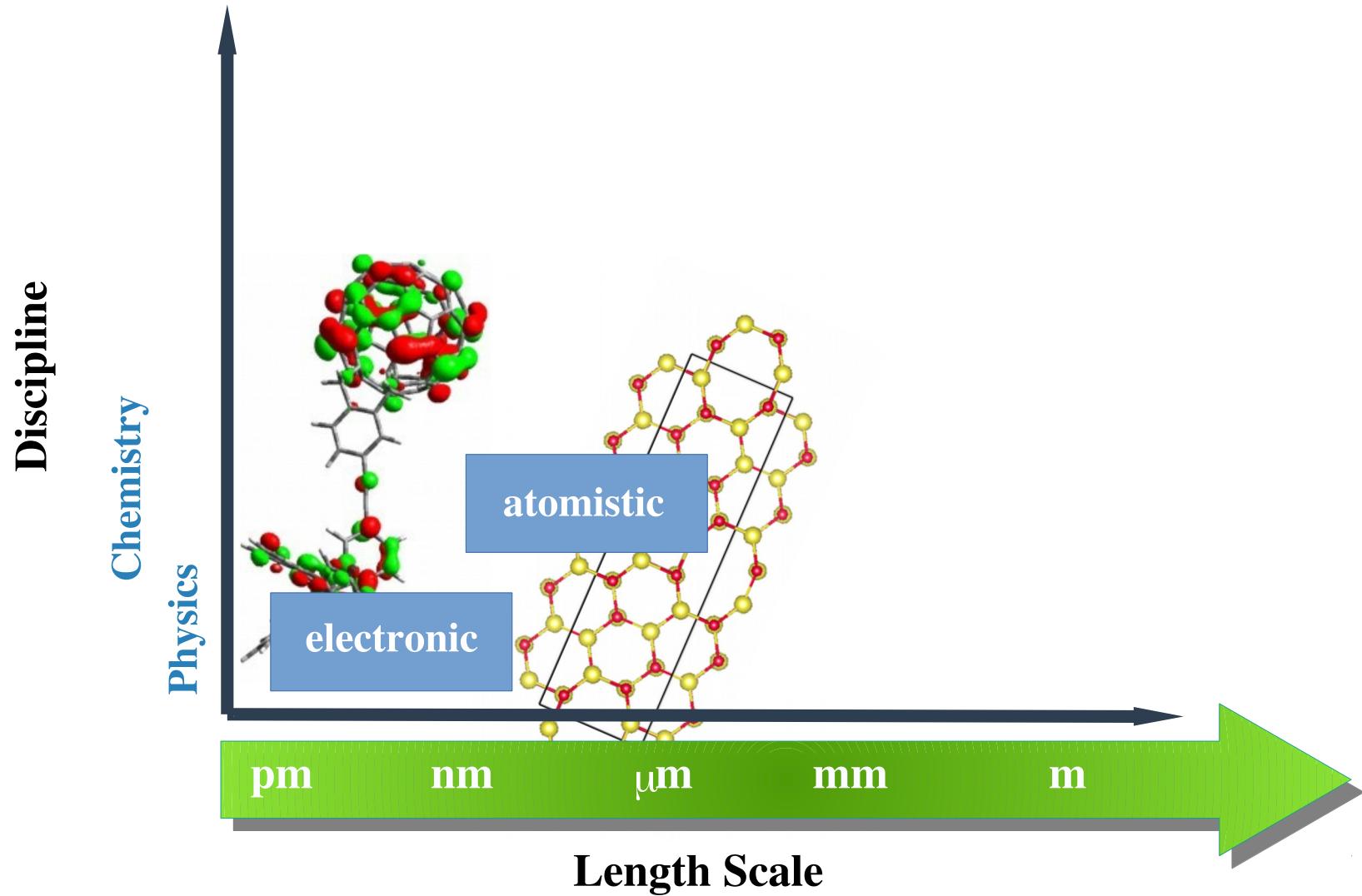
Institute for Advanced Studies
in Basic Sciences
Gava Zang, Zanjan, Iran

Condensed Matter

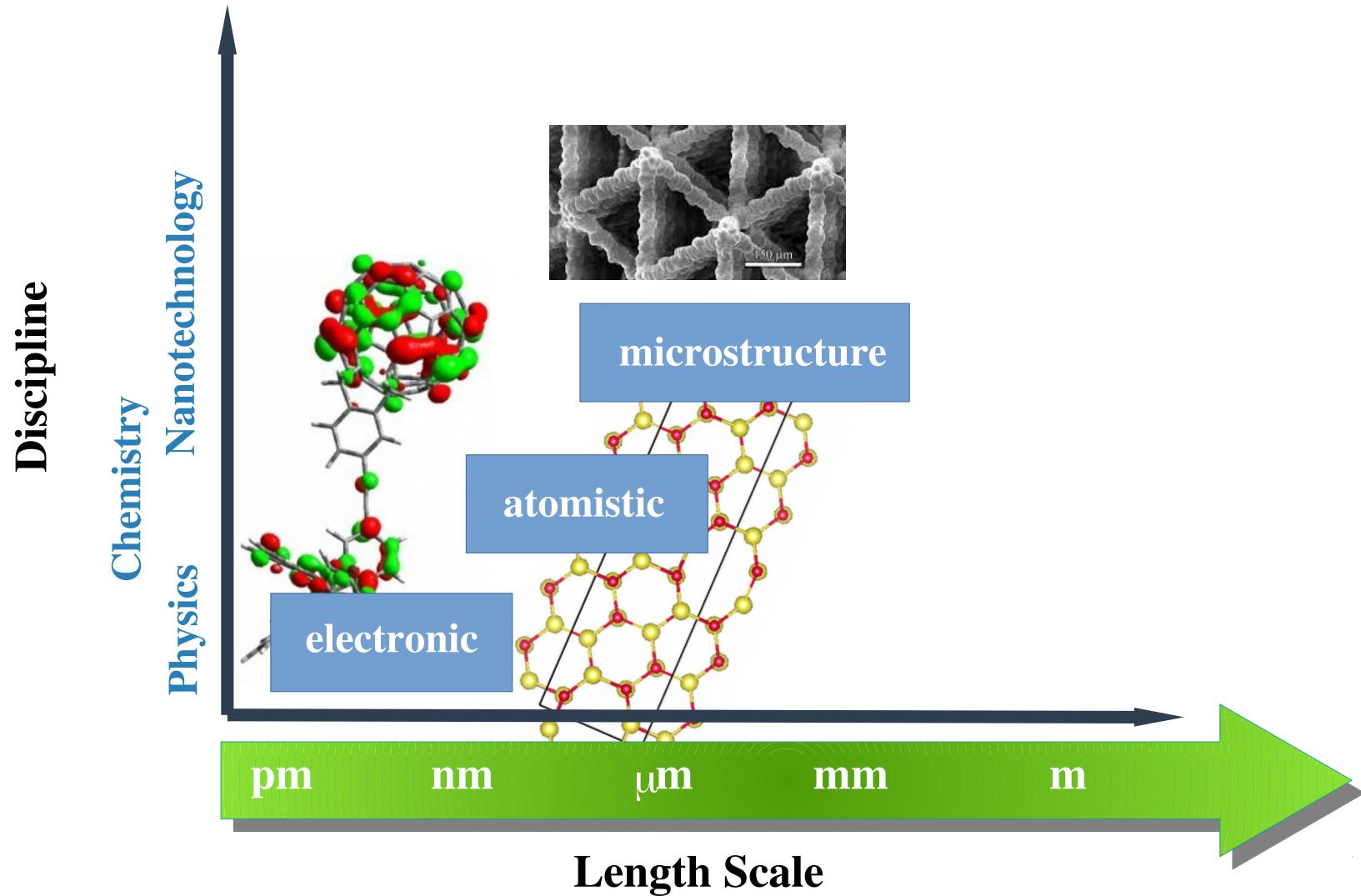
Condensed Matter



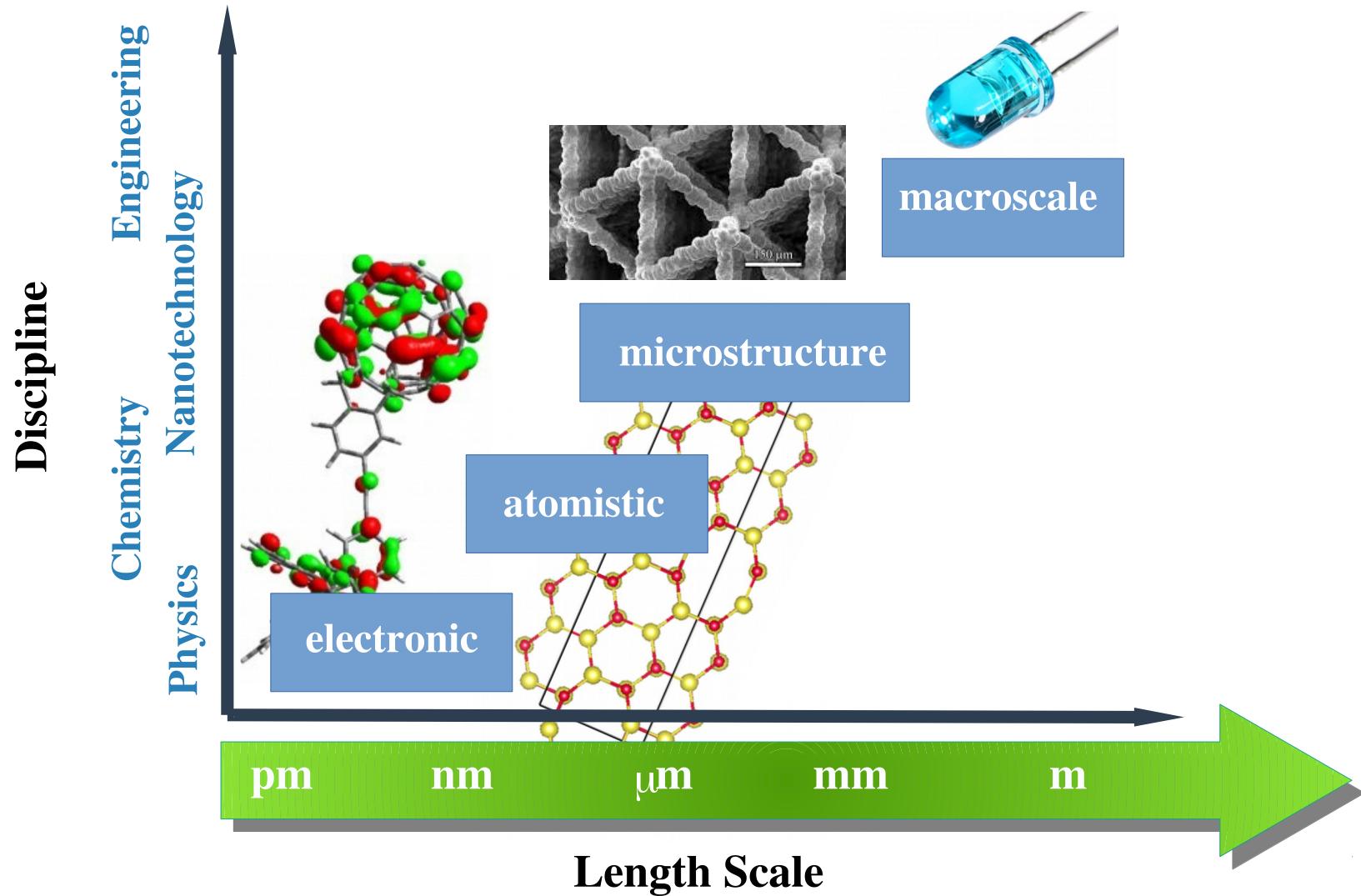
Condensed Matter



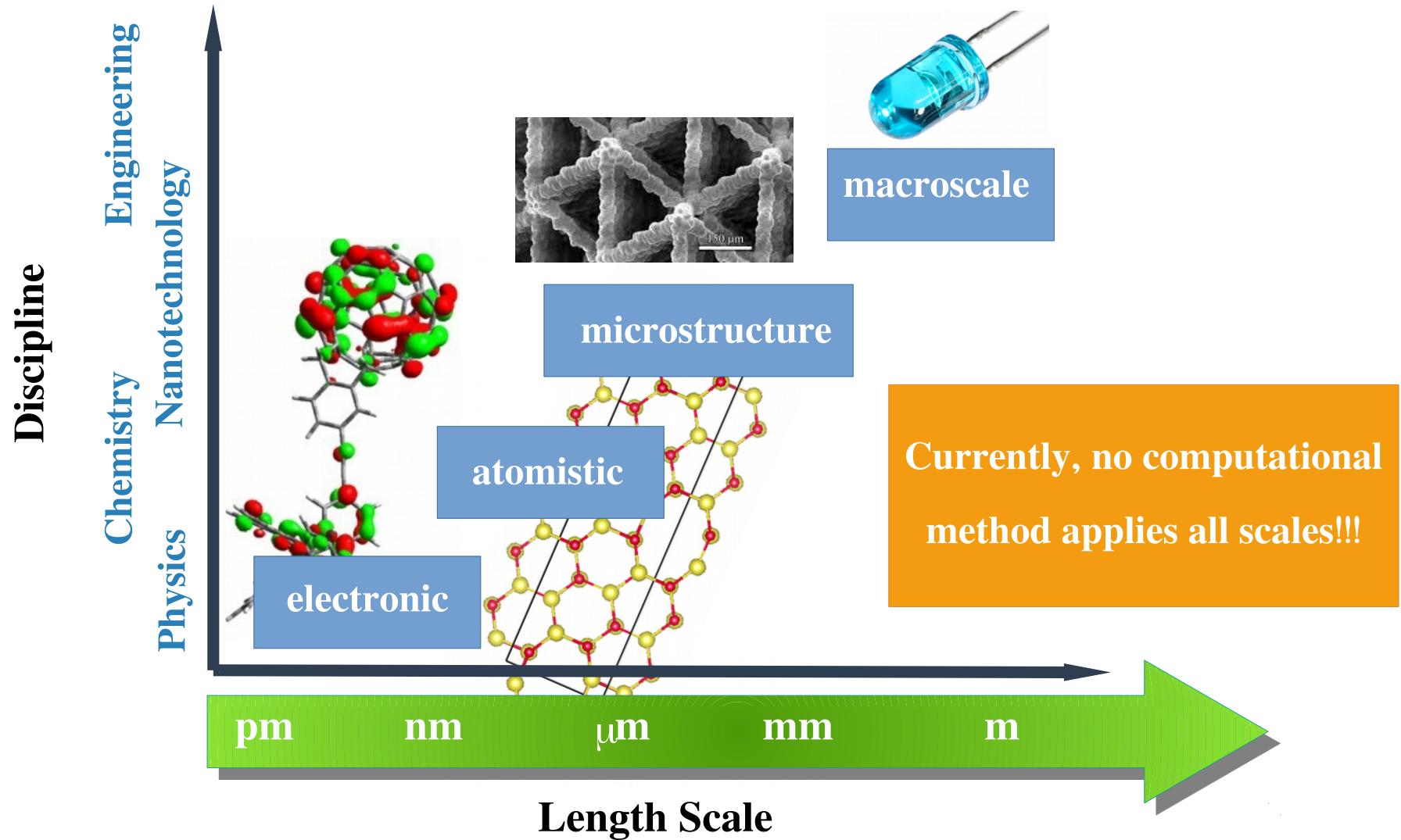
Condensed Matter



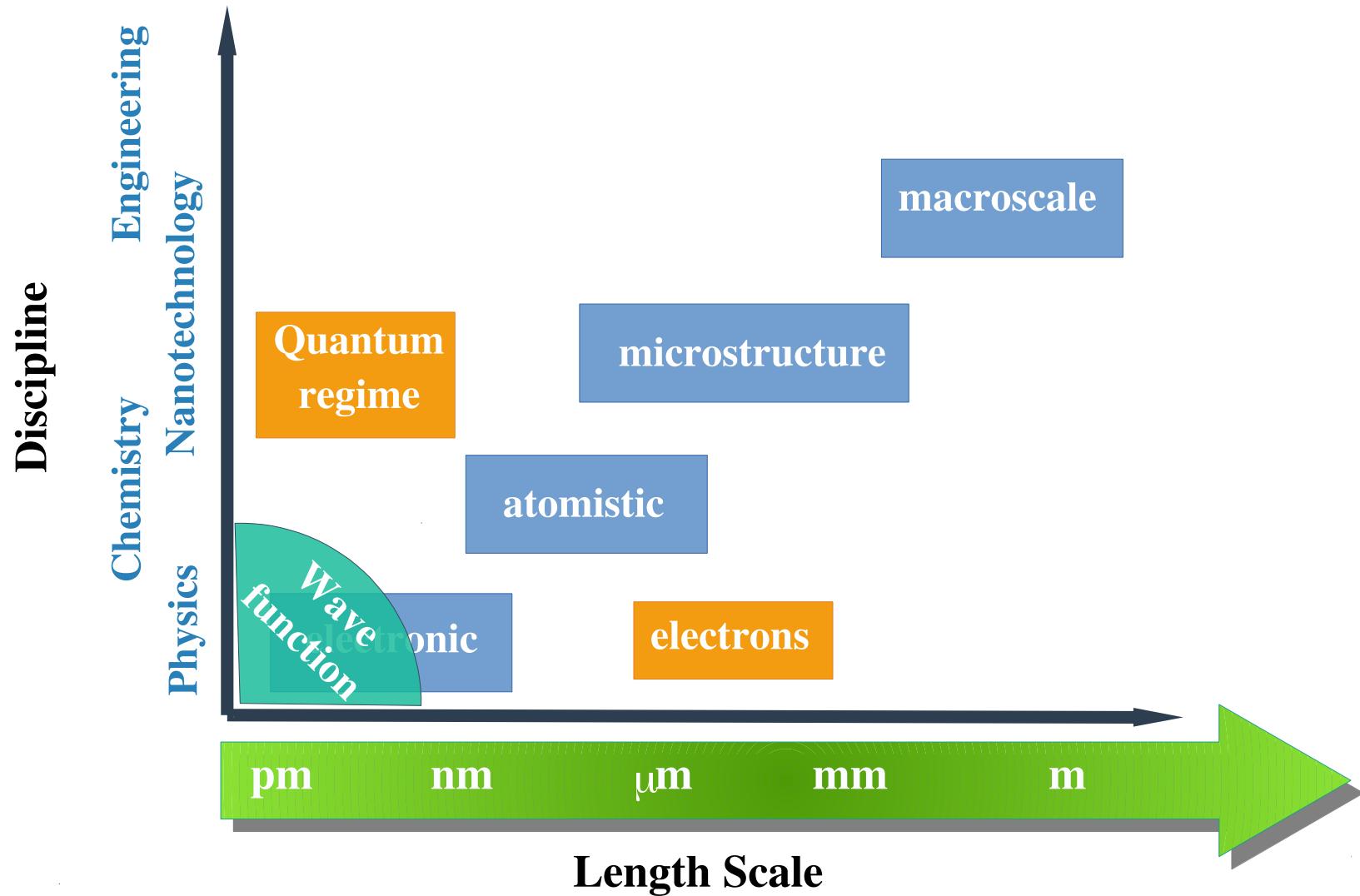
Condensed Matter



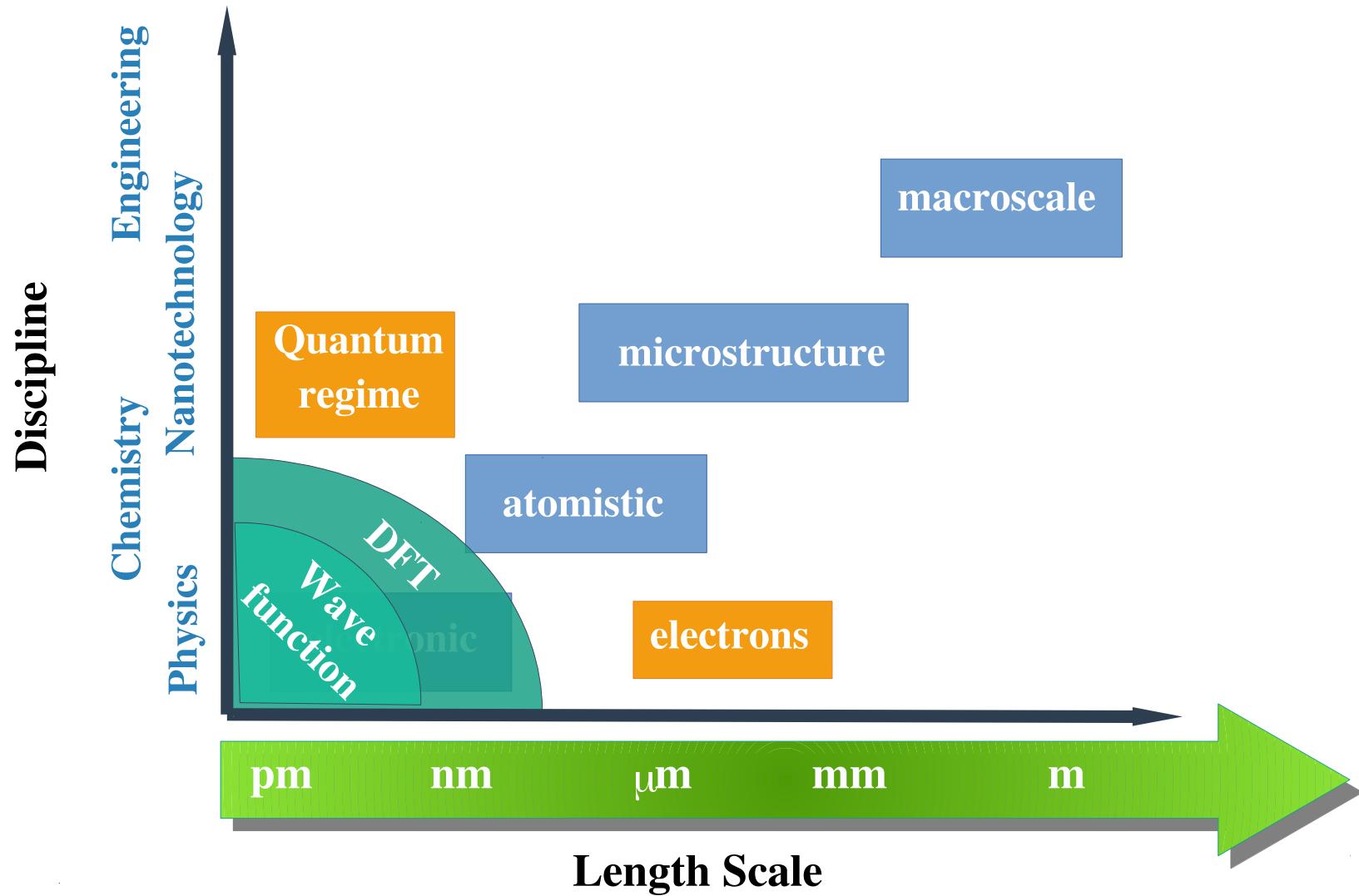
Condensed Matter



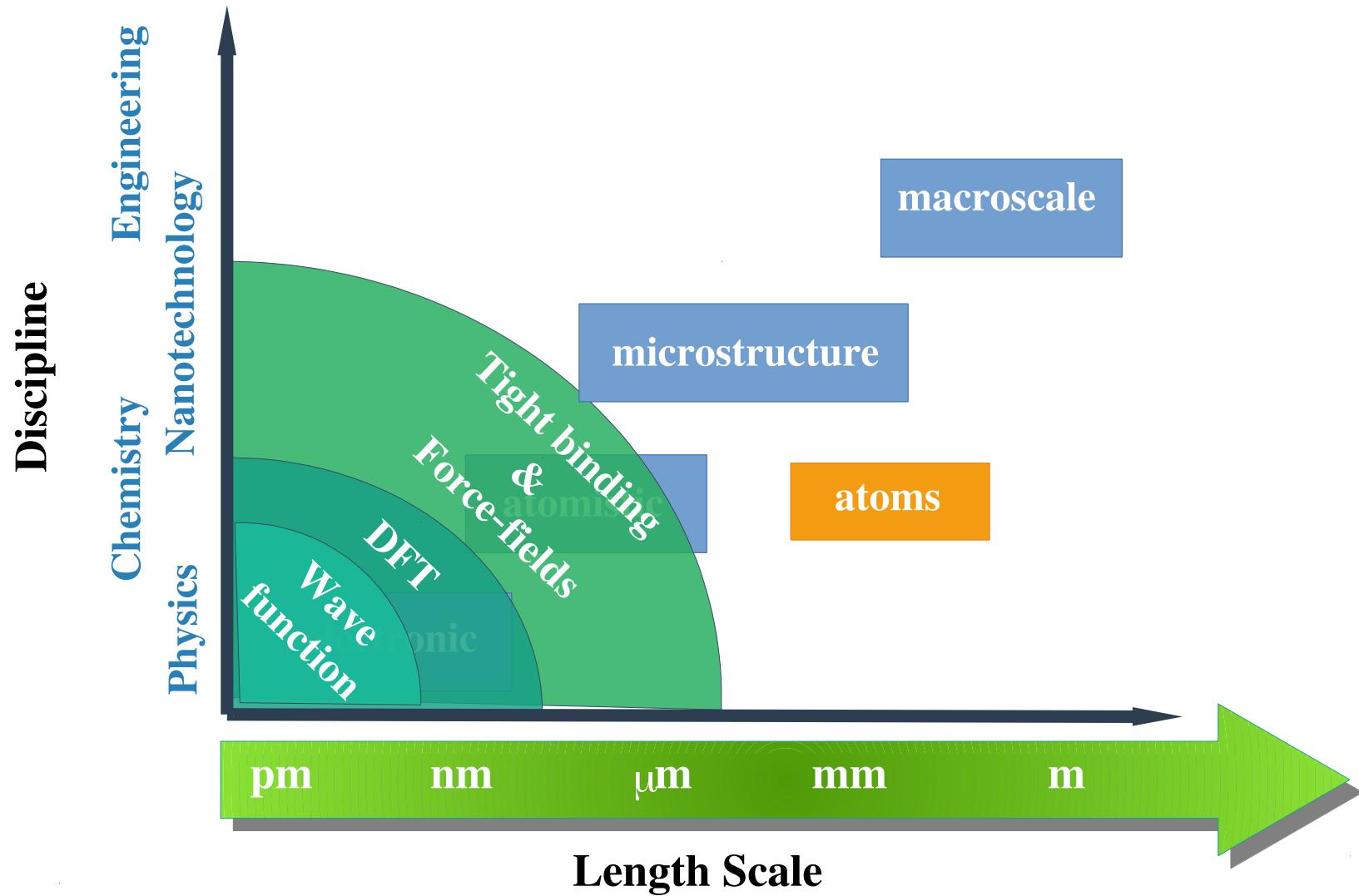
Hierarchy of Methods



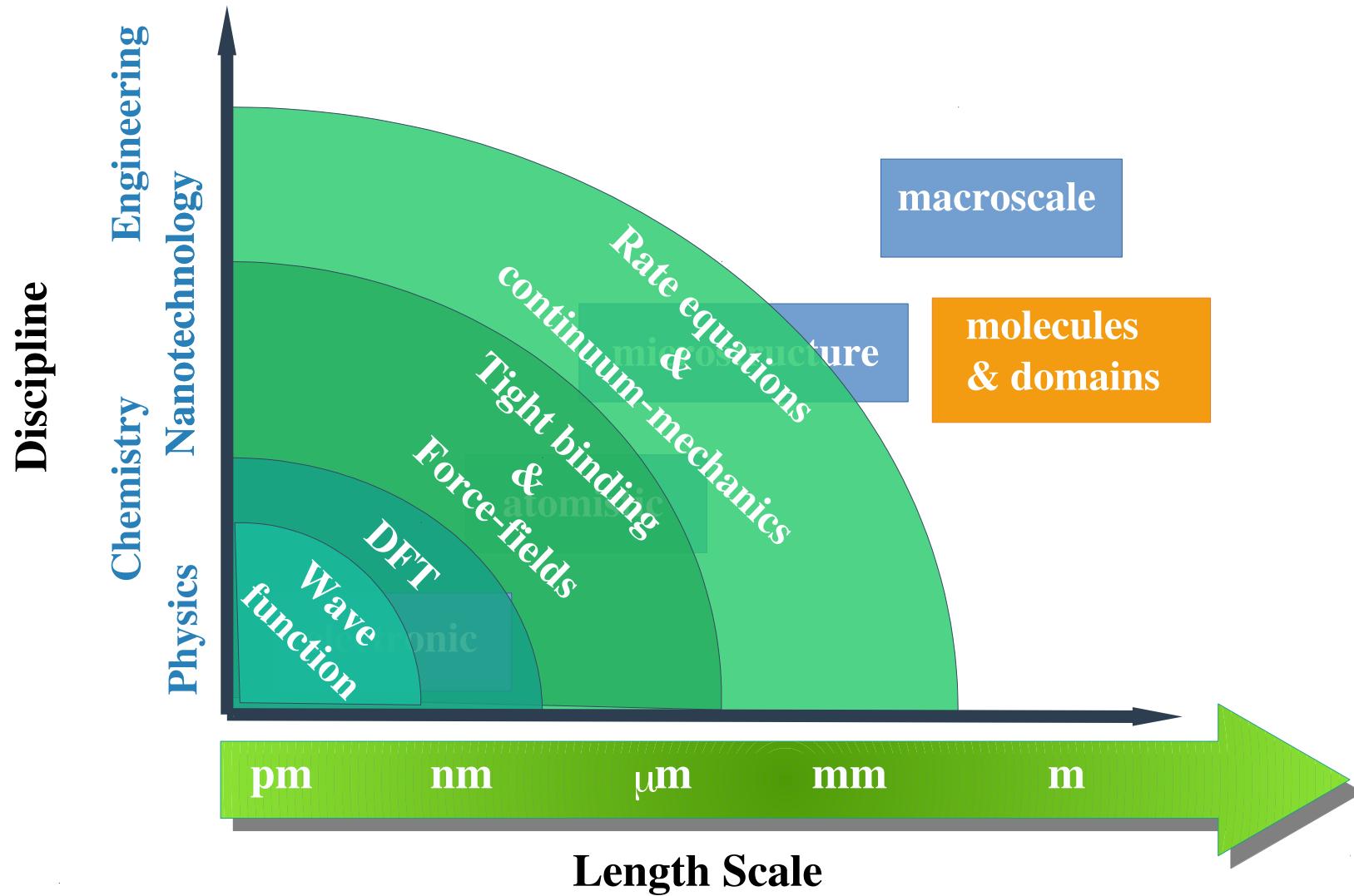
Hierarchy of Methods



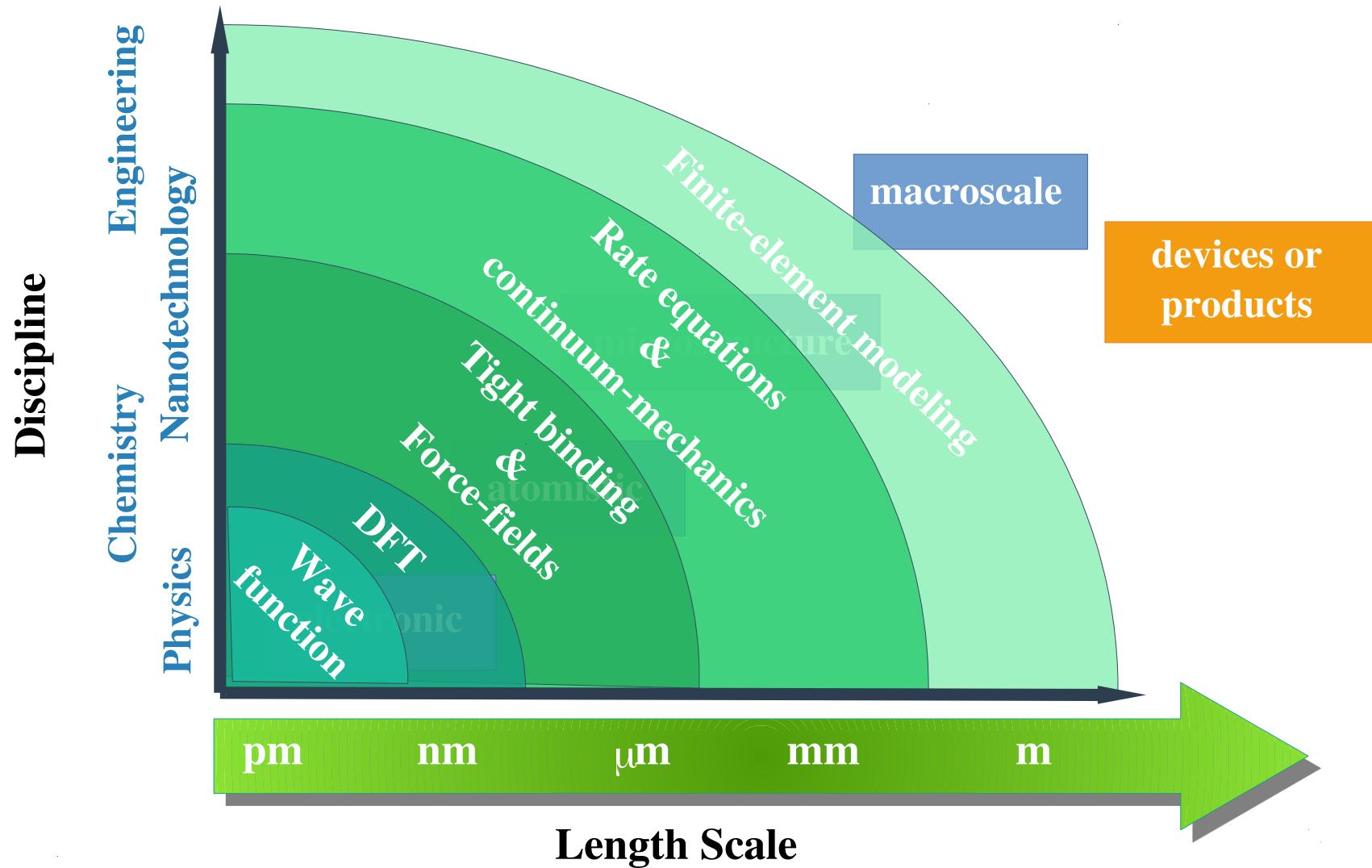
Hierarchy of Methods



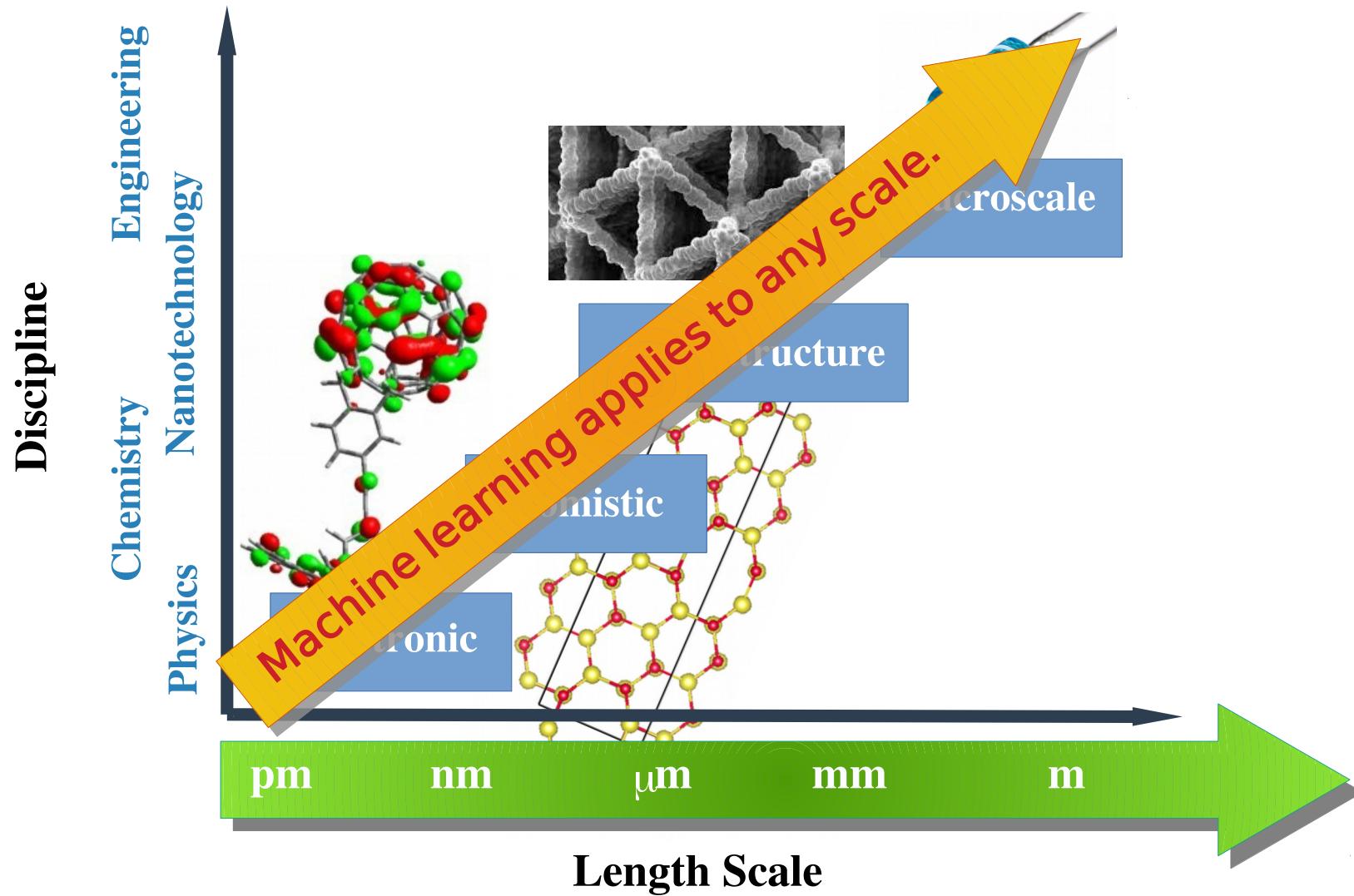
Hierarchy of Methods



Hierarchy of Methods



Hierarchy of Methods

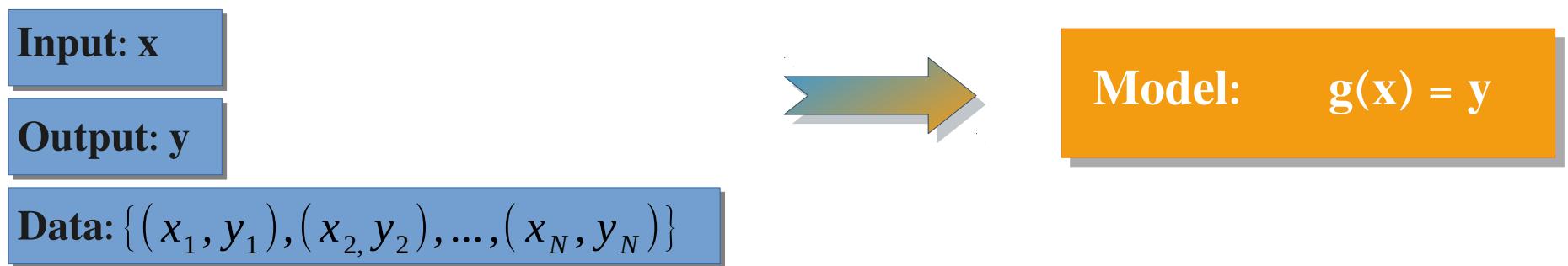


What is Machine Learning?

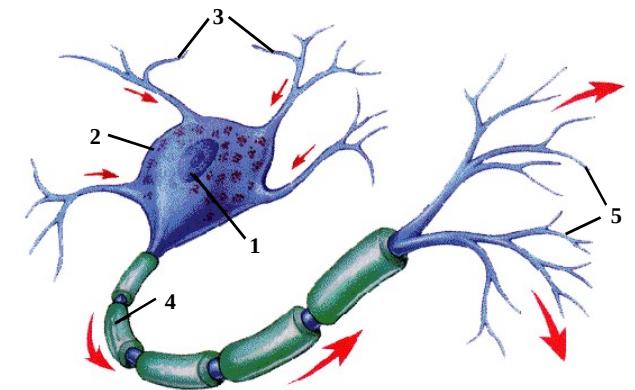
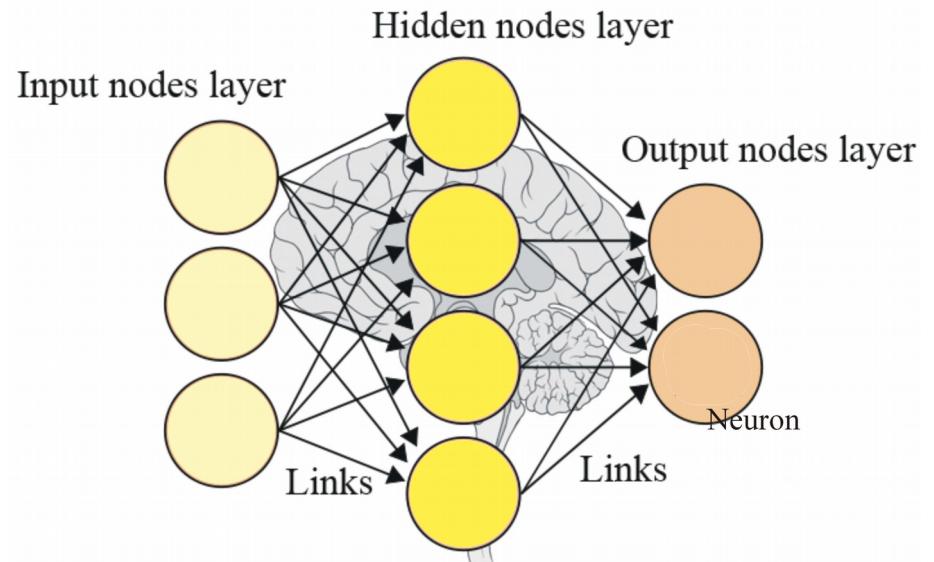
- **The context:** the ability of a learning machine to perform accurately on new, unseen experience/data after having experienced a learning data set.
- **The goal:** understand the structure of data and fit that data into models that can be understood and utilized by people.

What is Machine Learning?

- **The context:** the ability of a learning machine to perform accurately on new, unseen experience/data after having experienced a learning data set.
- **The goal:** understand the structure of data and fit that data into models that can be understood and utilized by people.



Artificial Neural Network (ANN)



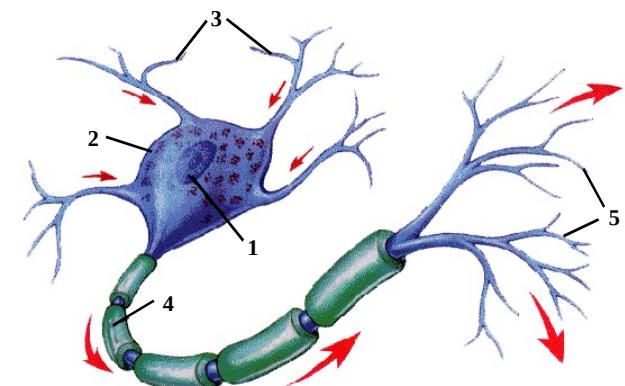
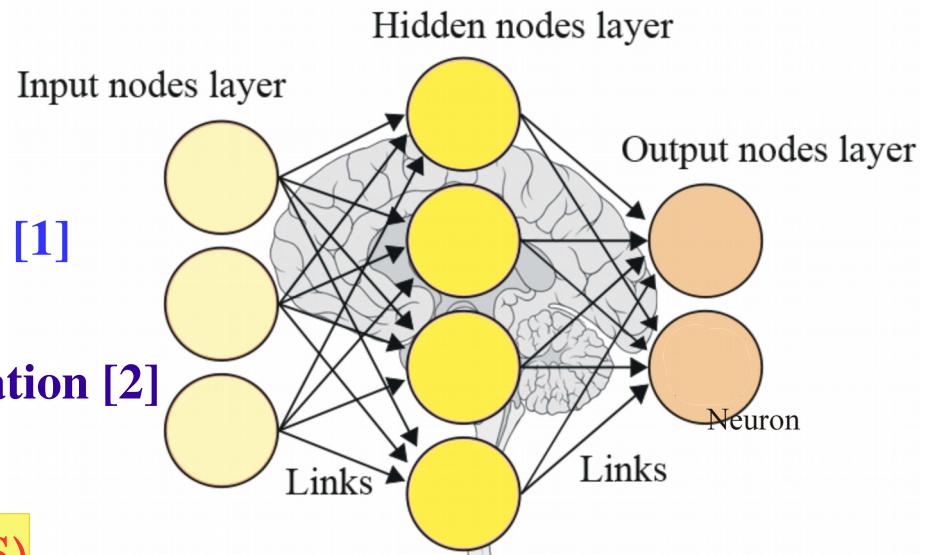
W. McCulloch, and W. Pitts, Bull. Math. Biophys. 5, 115, (1943)

Artificial Neural Network (ANN)

➤ Construction of exchange correlation potentials [1]

➤ The numerical solution of the Schrodinger equation [2]

➤ Construction of potential-energy surfaces (PES)



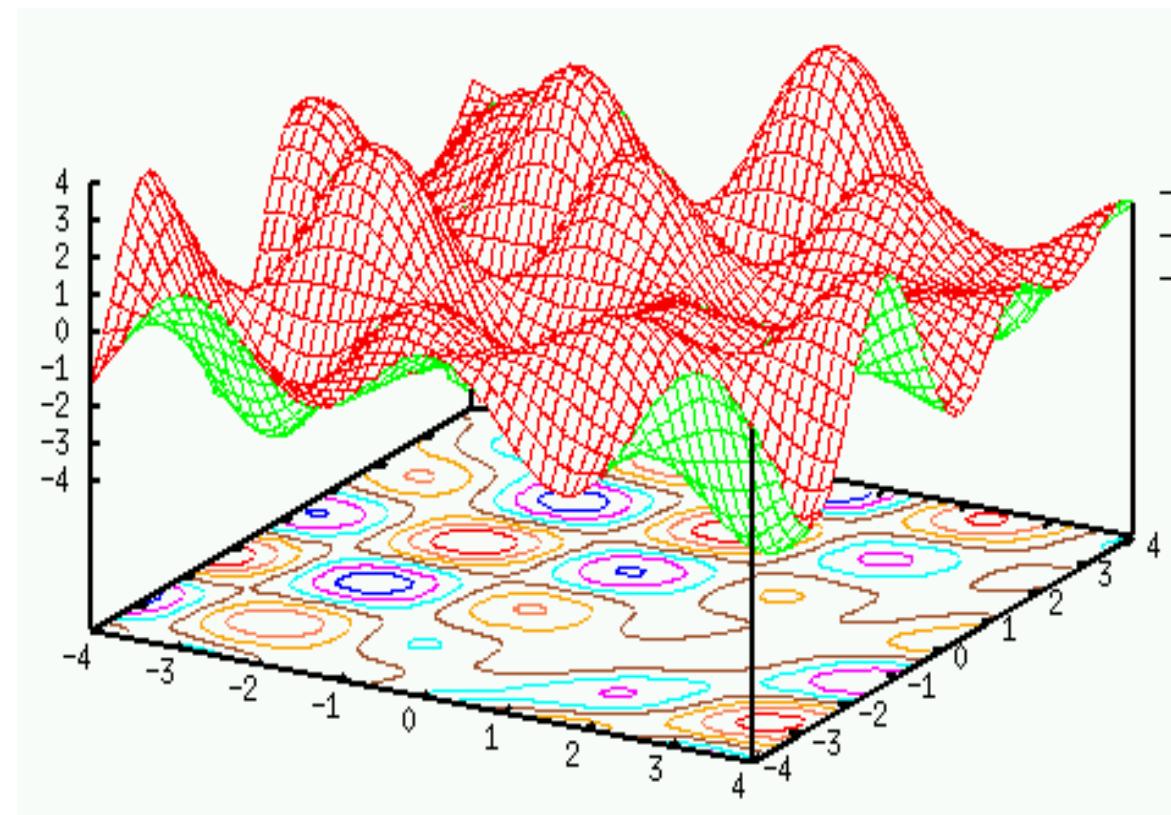
W. McCulloch, and W. Pitts, Bull. Math. Biophys. 5, 115, (1943)

[1] J. Gasteiger, and J. Zupan, Angew. Chem. 105, 510, (1993)

[2] X. Zheng et. al, Chen, Chem. Phys. Lett. 390, 186, (2004)

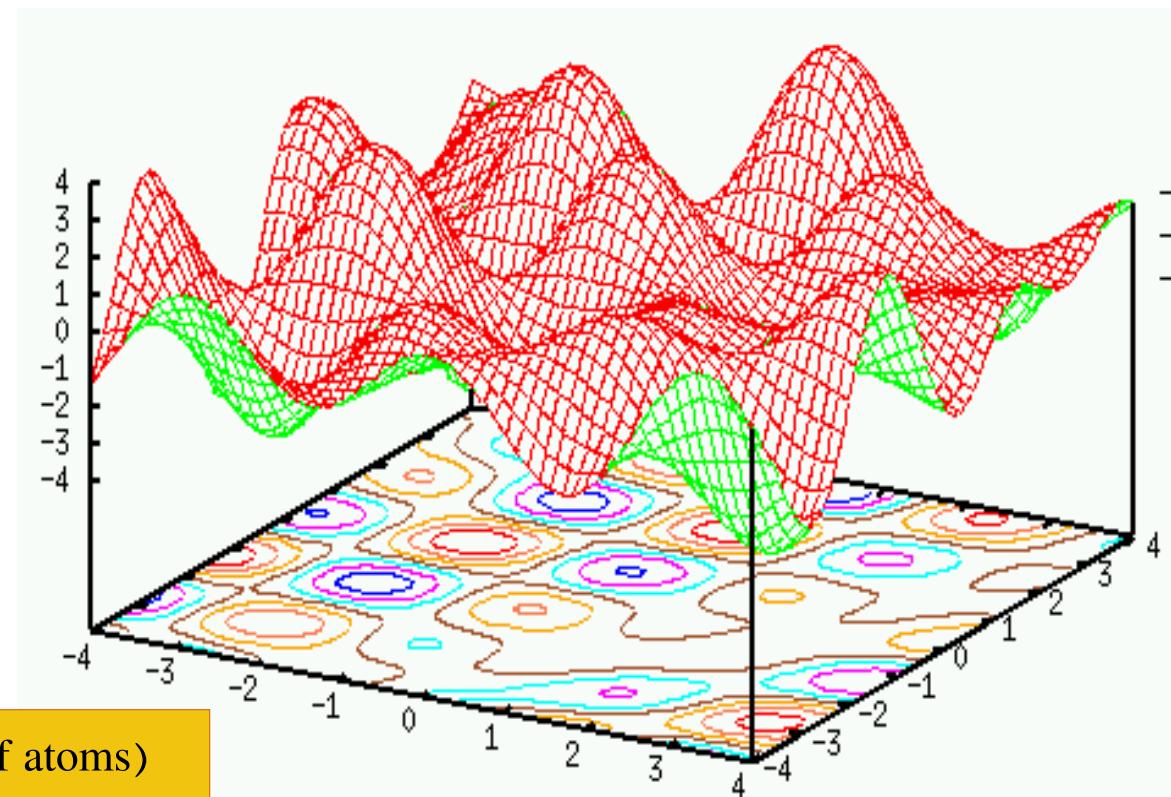
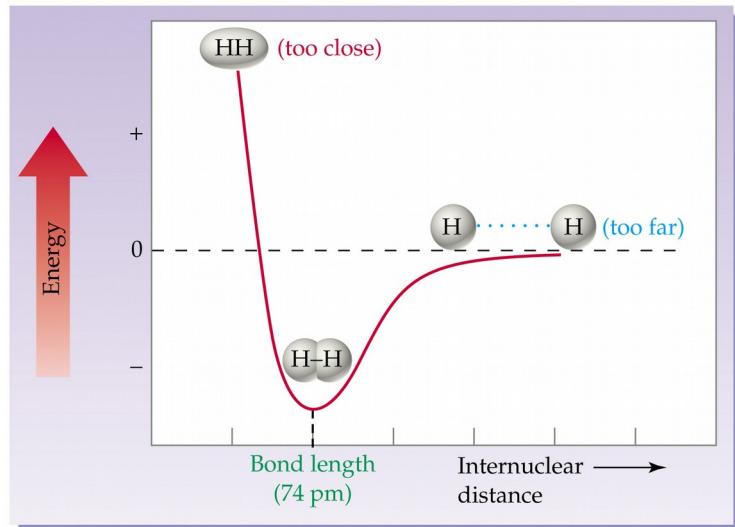
Potential Energy Surface (PES)

The total energy of the electronic ground state as a function of coordinates of the atoms in a molecule or solid $E(R_1, \dots, R_N)$ is called the potential energy surface (PES).



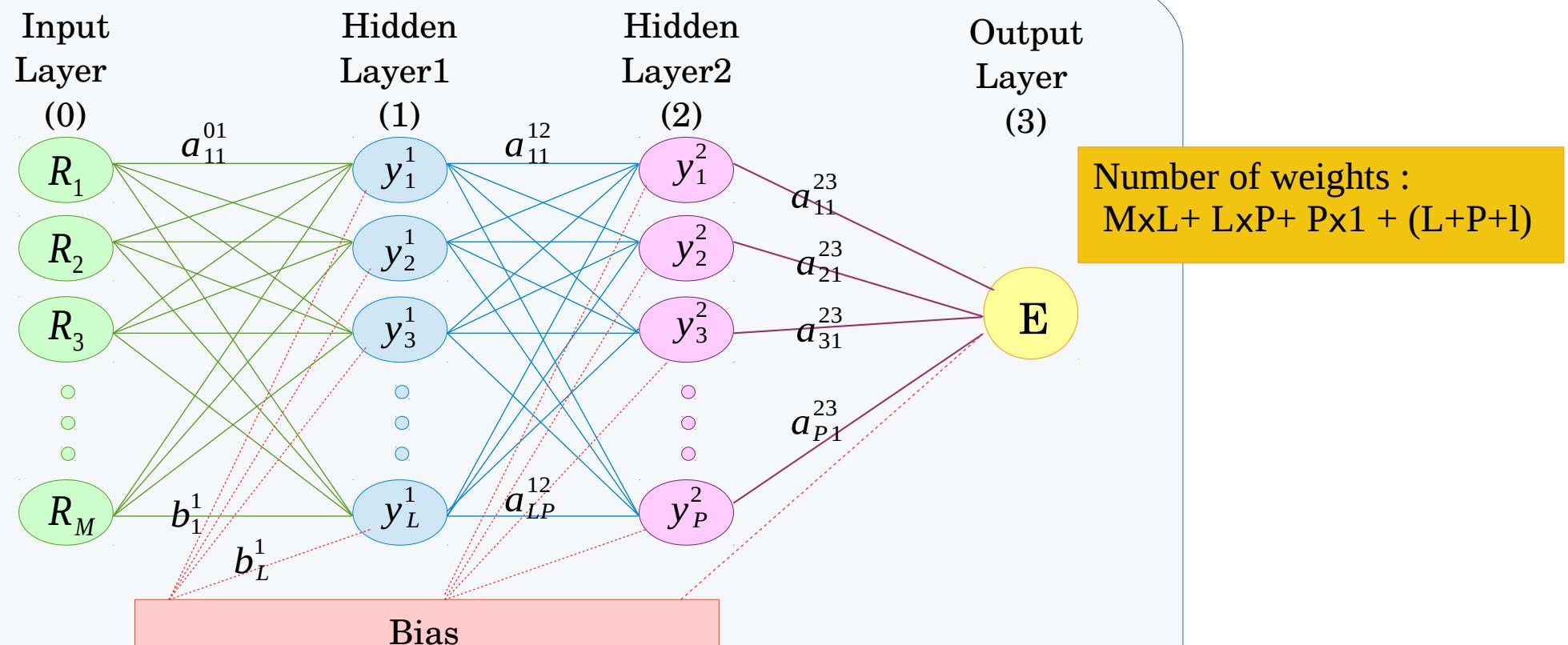
Potential Energy Surface (PES)

The total energy of the electronic ground state as a function of coordinates of the atoms in a molecule or solid $E(R_1, \dots, R_N)$ is called the potential energy surface (PES).



PES tells us which structures (arrangement of atoms) are energetically favorable (low in energy).

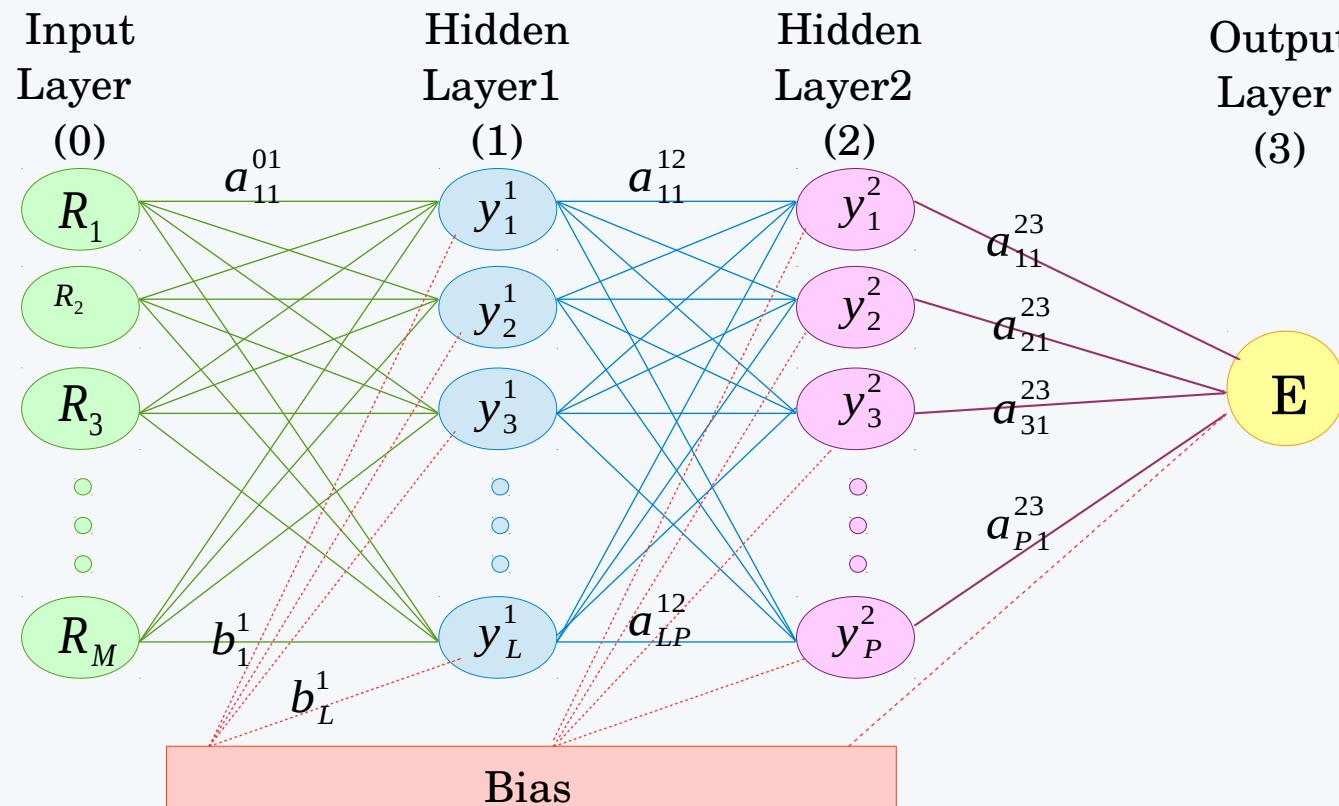
Construction of PES with ANN



Schematic structure of a simple M-L-P-1 feed-forward neural network.

S. Lorenz, A. Groß, and M. Scheffler, Chemical Physics Letters 395, 210 (2004).
J. Behler, S. Lorenz, and K. Reuter, J. Chem. Phys. 014705, 127 (2007).

Construction of PES with ANN



Number of weights :
 $M \times L + L \times P + P \times 1 + (L + P + 1)$

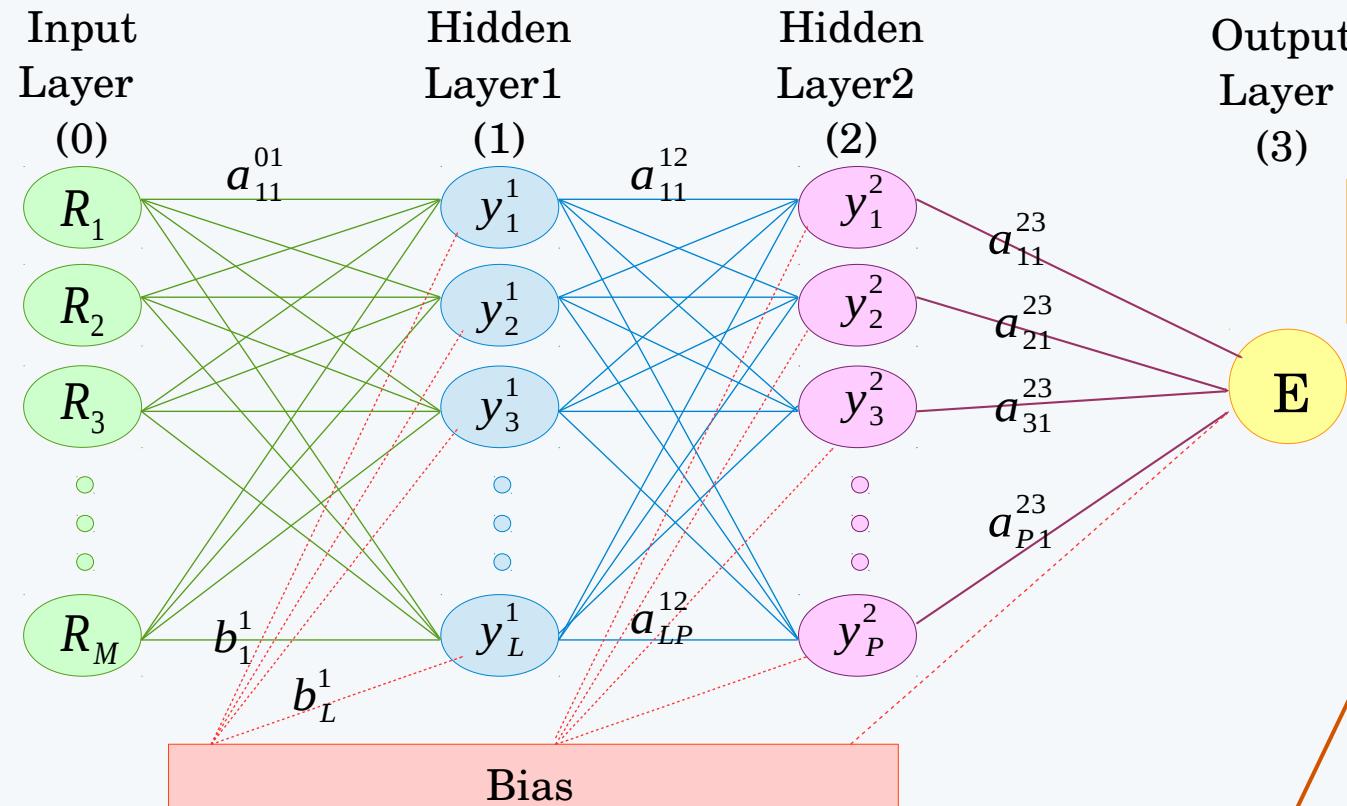
Activation functions

$$y_j^1 = f_j^1(x) = \tanh(x)$$

$$y_k^2 = f_k^2(x) = \tanh(x)$$

$$E = y_1^3 = f_1^3(x) = x$$

Construction of PES with ANN



Number of weights :
 $M \times L + L \times P + P \times 1 + (L + P + 1)$

Activation functions

$$y_j^1 = f_j^1(x) = \tanh(x)$$

$$y_k^2 = f_k^2(x) = \tanh(x)$$

$$E = y_1^3 = f_1^3(x) = x$$

$$E = f_1^3(b_1^3 + \sum_{k=1}^P a_{k1}^{23} \cdot f_k^2(b_k^2 + \sum_{j=1}^L a_{jk}^{12} \cdot f_j^1(b_j^1 + \sum_{i=1}^M a_{ij}^{01} \cdot R_i)))$$

Learning or Training Process

$$\chi = 1/N \sum_i (E_i^{ANN} - E_i^{ref})^2$$

Optimization algorithms:

- Steepest descent
- Levenberg-Marquardt
- Kalman-filter
- ...

Learning or Training Process

$$\chi = 1/N \sum_i (E_i^{ANN} - E_i^{ref})^2$$

Optimization algorithms:

- Steepest descent
- Levenberg-Marquardt
- Kalman-filter
- ...

Algorithm 2: Gradient Descent

input : $f : \mathbb{R}^n \rightarrow \mathbb{R}$ a differentiable function
 $\mathbf{x}^{(0)}$ an initial solution
output: \mathbf{x}^* , a local minimum of the cost function f .

```
1 begin
2    $k \leftarrow 0$  ;
3   while STOP-CRIT and ( $k < k_{max}$ ) do
4      $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} - \alpha^{(k)} \nabla f(\mathbf{x})$  ;
5     with  $\alpha^{(k)} = \arg \min_{\alpha \in \mathbb{R}_+} f(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}))$  ;
6      $k \leftarrow k + 1$  ;
7   return  $\mathbf{x}^{(k)}$ 
8 end
```

Learning or Training Process

$$\chi = 1/N \sum_i (E_i^{ANN} - E_i^{ref})^2$$

Optimization algorithms:

- Steepest descent
- Levenberg-Marquardt
- Kalman-filter
- ...

Algorithm 5: Levenberg-Marquardt algorithm

input : $f : \mathbb{R}^n \rightarrow \mathbb{R}$ a function such that $f(\mathbf{x}) = \sum_{i=1}^m (f_i(\mathbf{x}))^2$
where all the f_i are differentiable functions from \mathbb{R}^n to \mathbb{R}
 $\mathbf{x}^{(0)}$ an initial solution

output: \mathbf{x}^* , a local minimum of the cost function f .

begin

 2 $k \leftarrow 0$;

 3 $\lambda \leftarrow \max \text{diag}(\mathbf{J}^T \mathbf{J})$;

 4 $\mathbf{x} \leftarrow \mathbf{x}^{(0)}$;

 5 **while** STOP-CRIT **and** ($k < k_{max}$) **do**

 6 Find δ such that $(\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J}))\delta = \mathbf{J}^T \mathbf{f}$;

 7 $\mathbf{x}' \leftarrow \mathbf{x} + \delta$;

 8 **if** $f(\mathbf{x}') < f(\mathbf{x})$ **then**

 9 $\mathbf{x} \leftarrow \mathbf{x}'$;

 10 $\lambda \leftarrow \frac{\lambda}{\nu}$;

 11 **else**

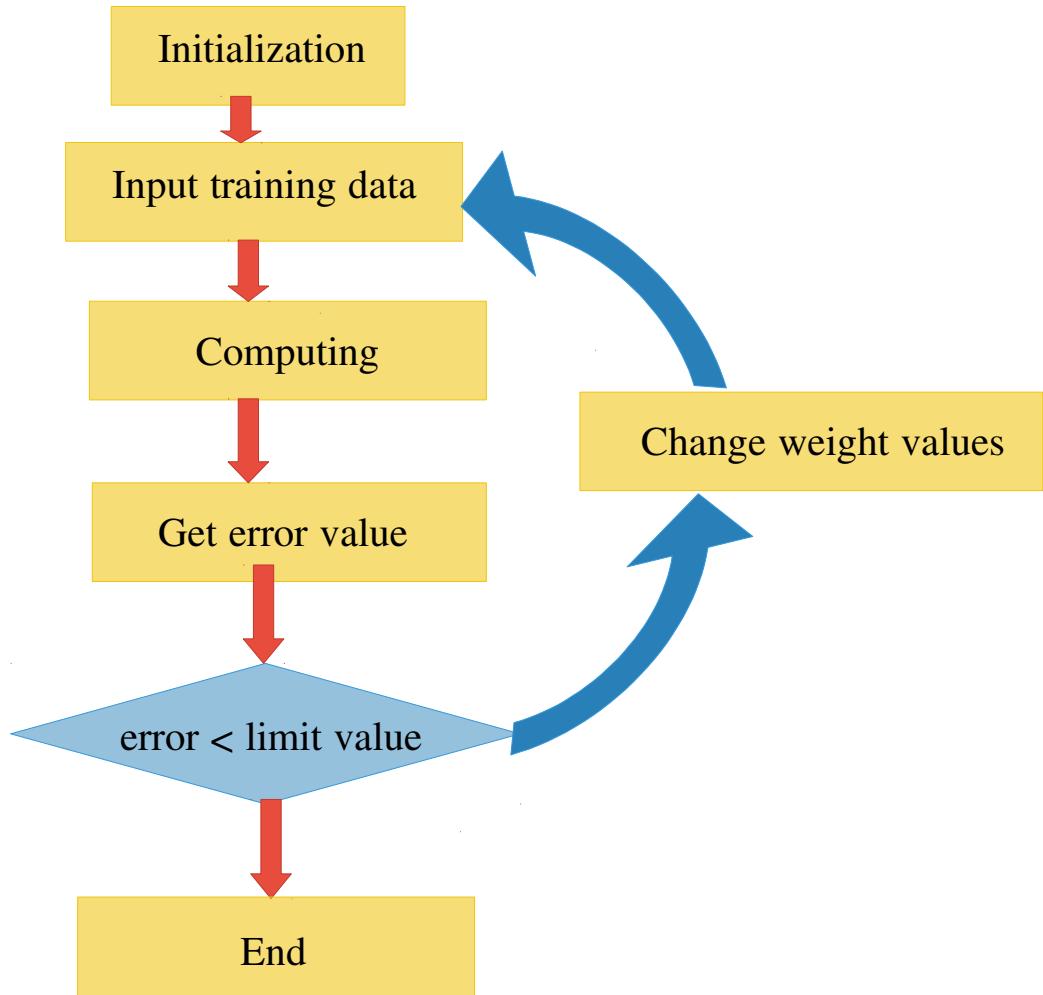
 12 $\lambda \leftarrow \nu \lambda$;

 13 $k \leftarrow k + 1$;

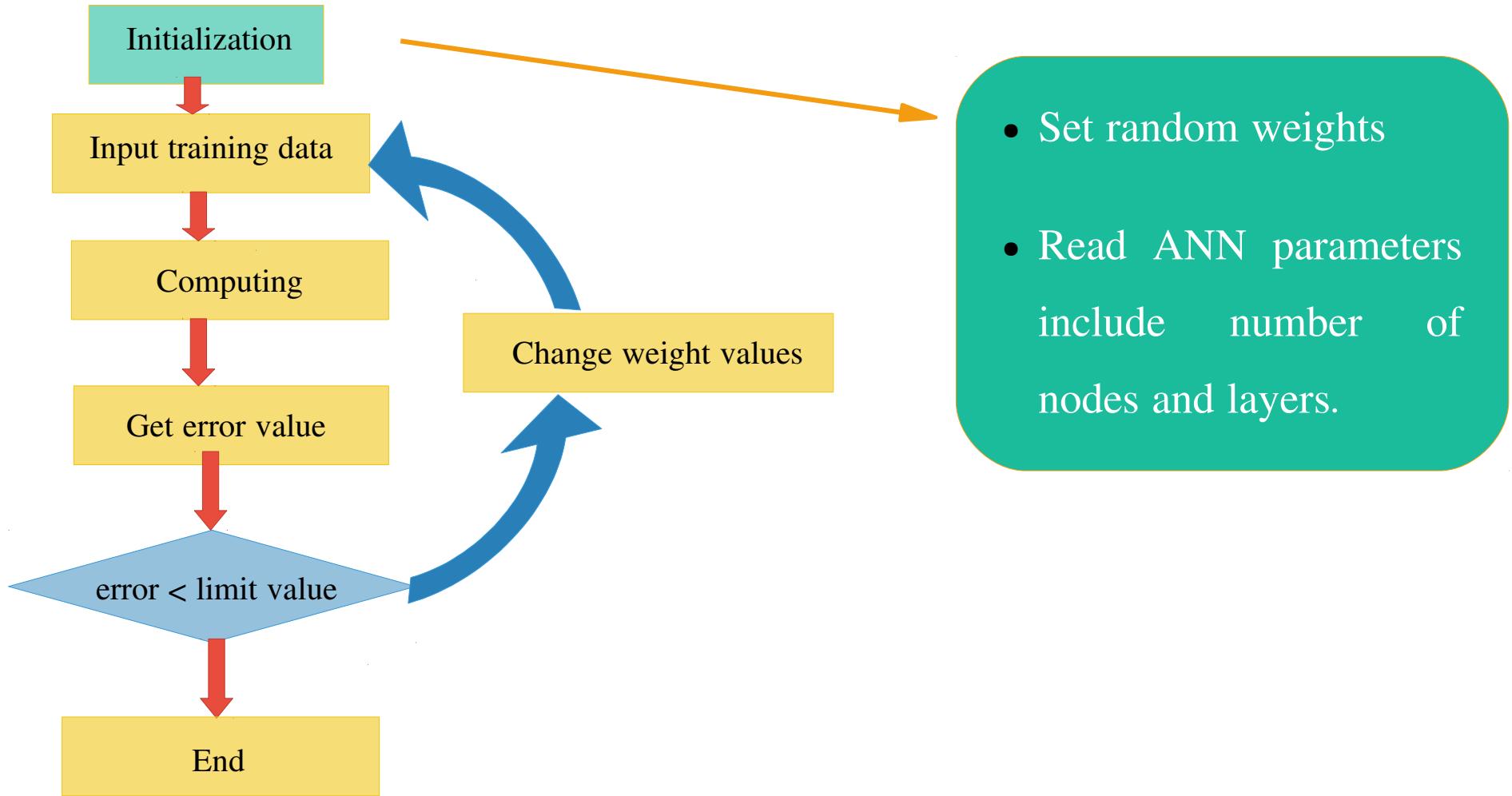
 14 **return** \mathbf{x}

 15 **end**

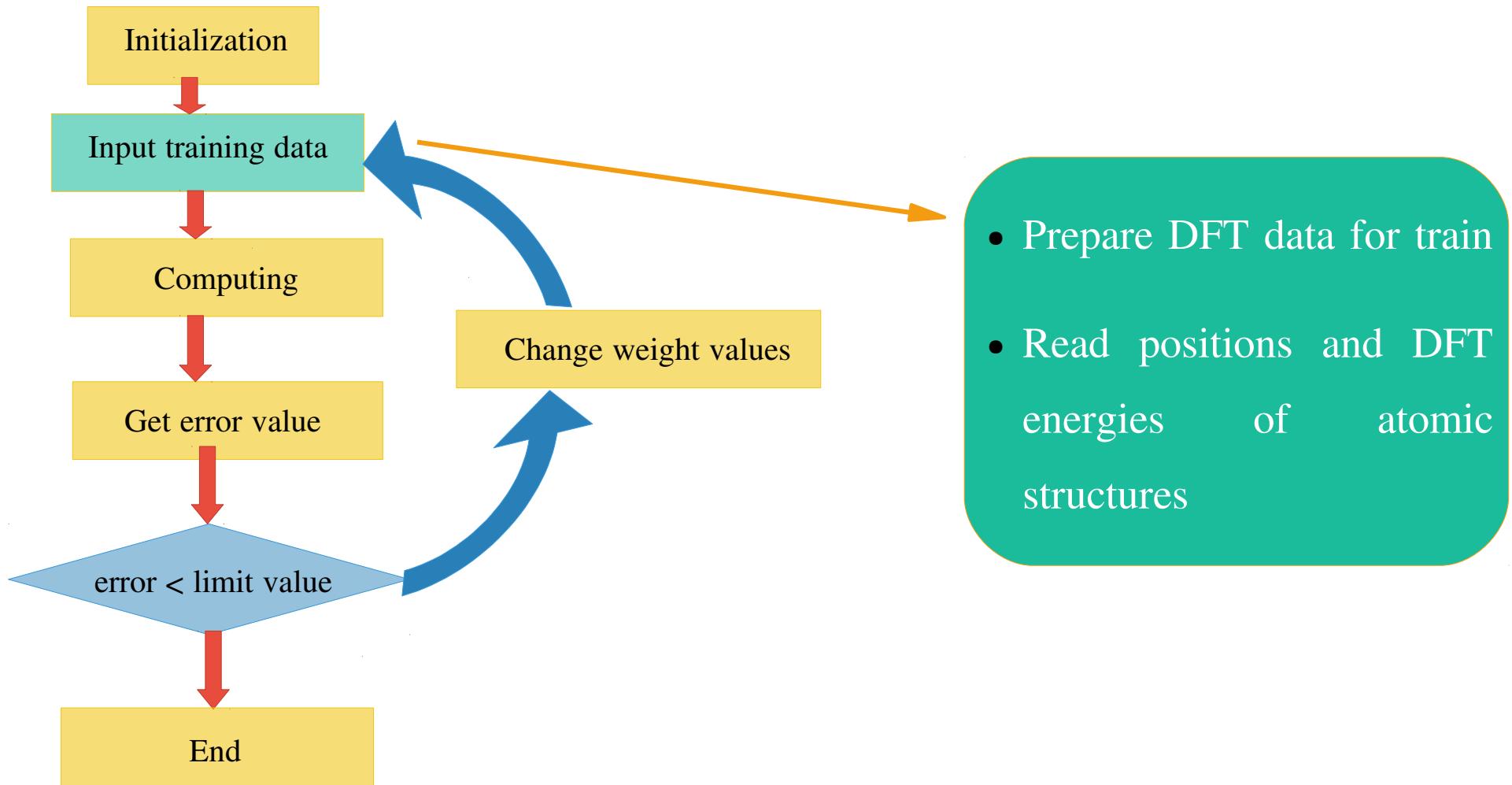
Training Flowchart



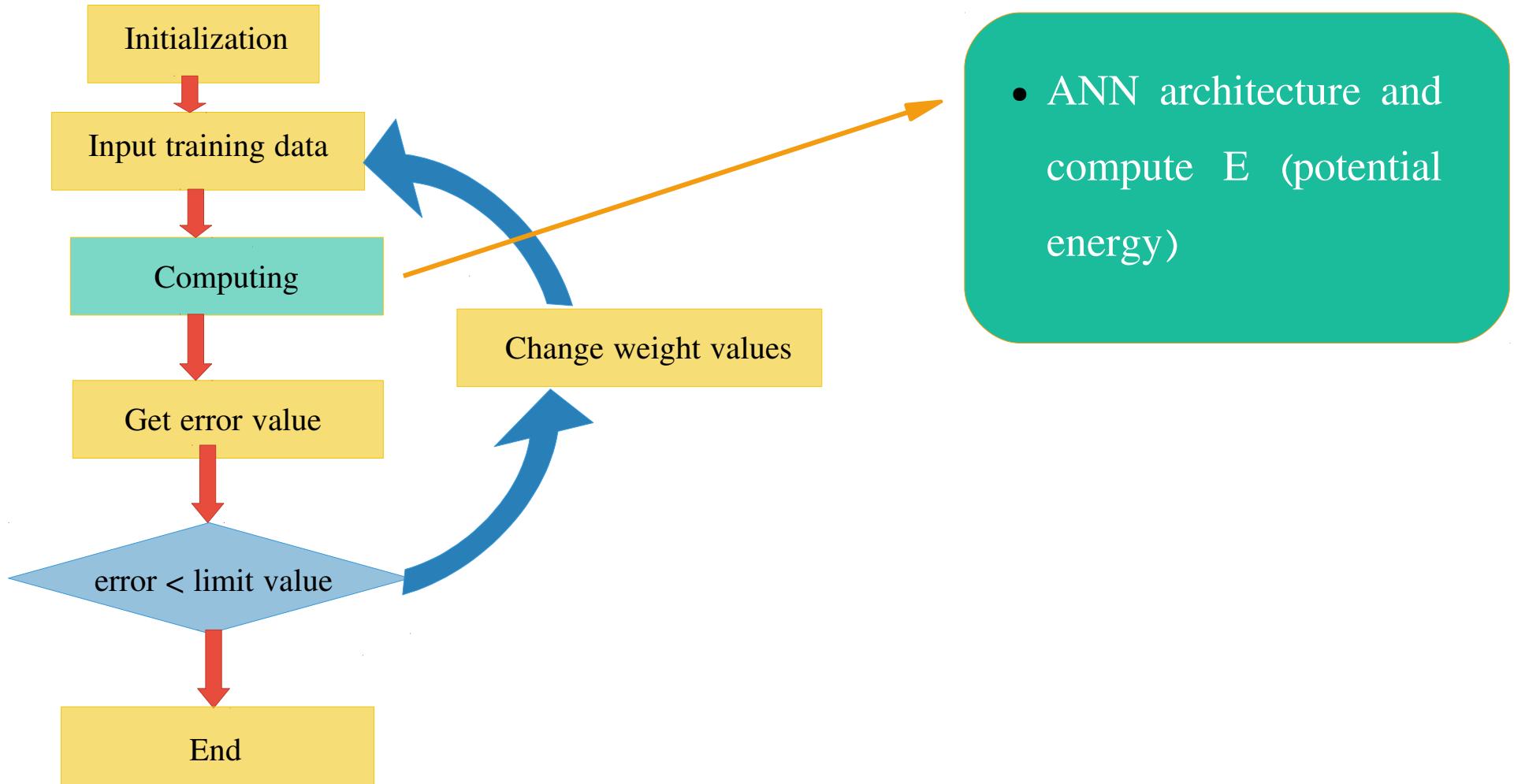
Training Flowchart



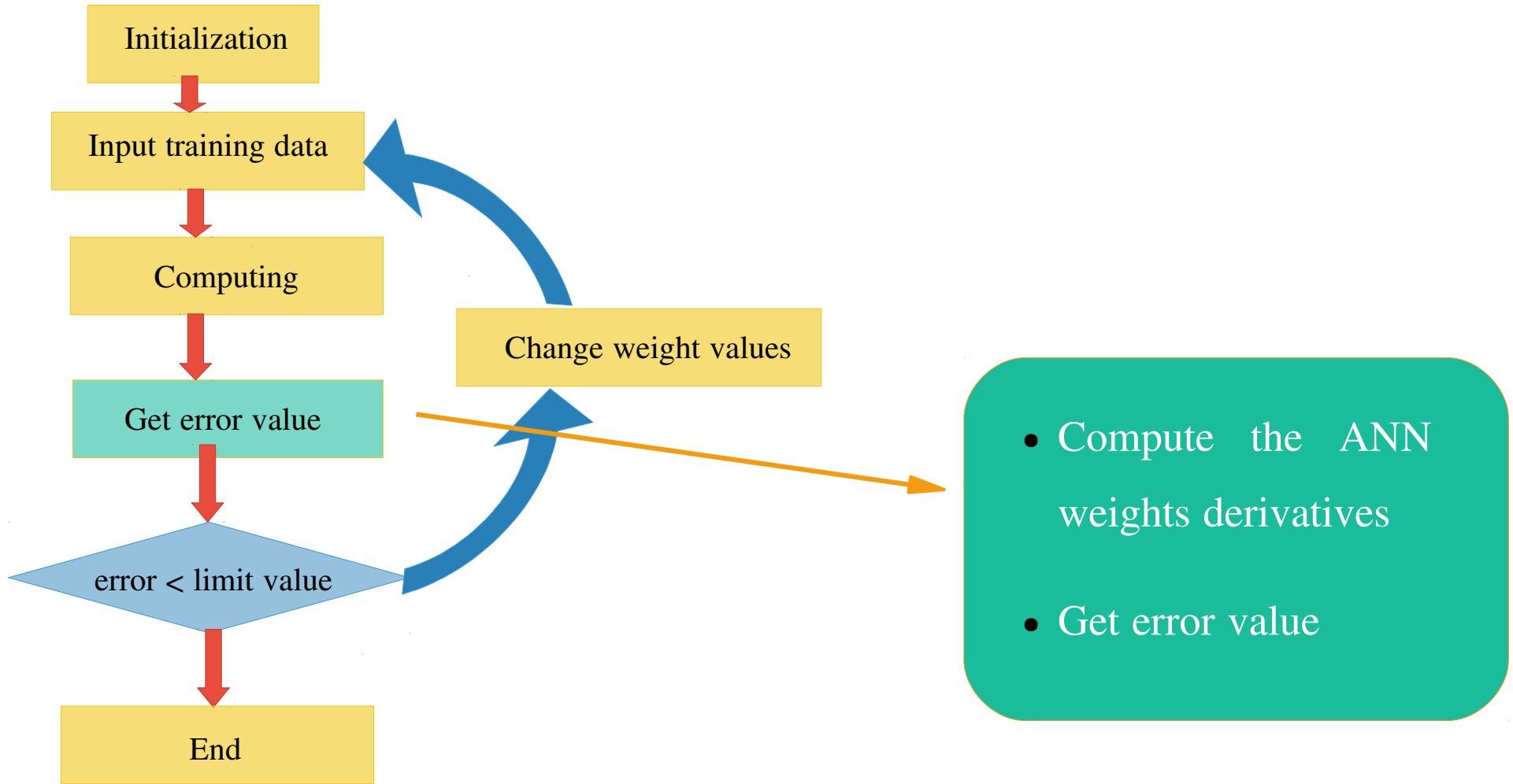
Training Flowchart



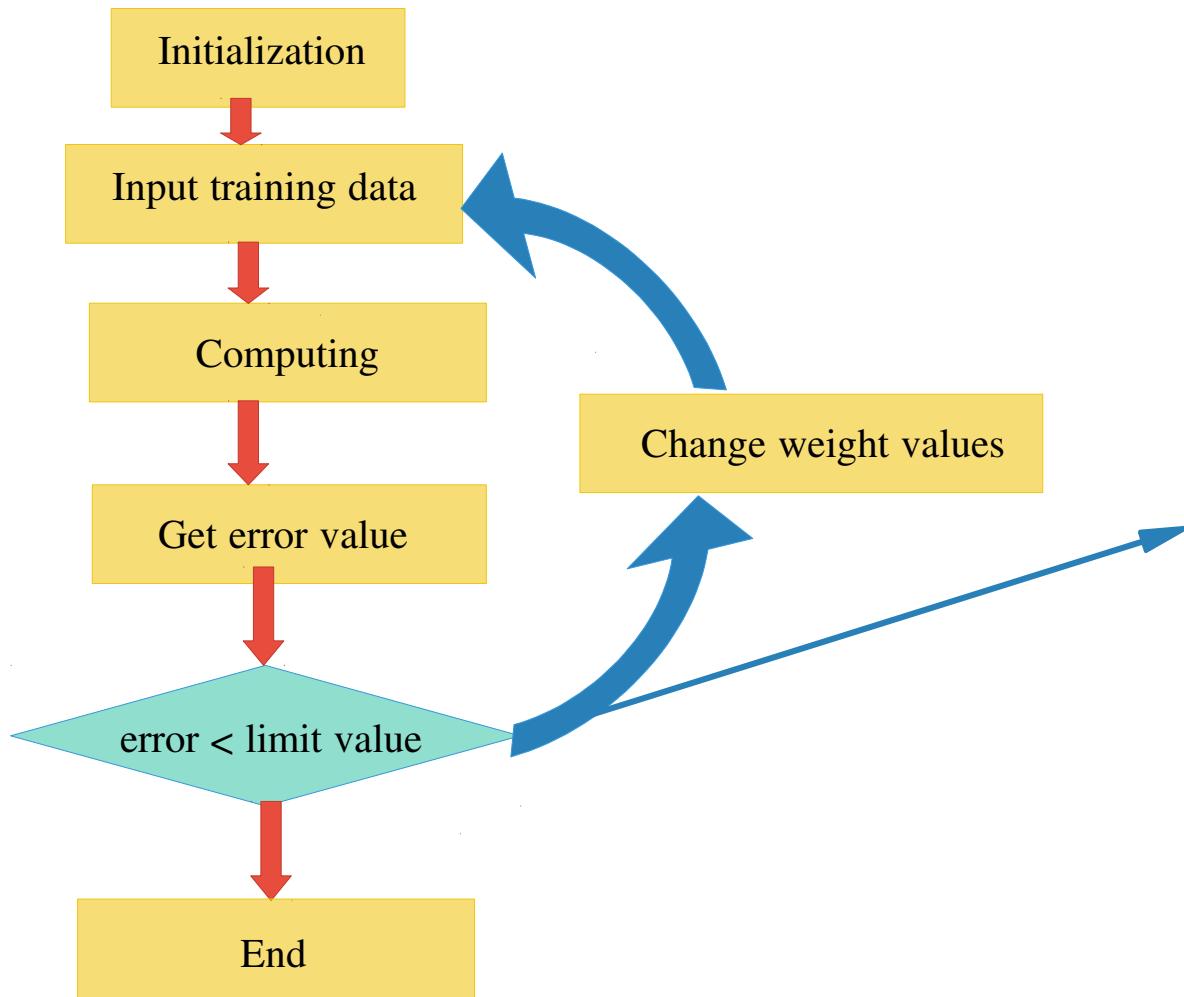
Training Flowchart



Training Flowchart

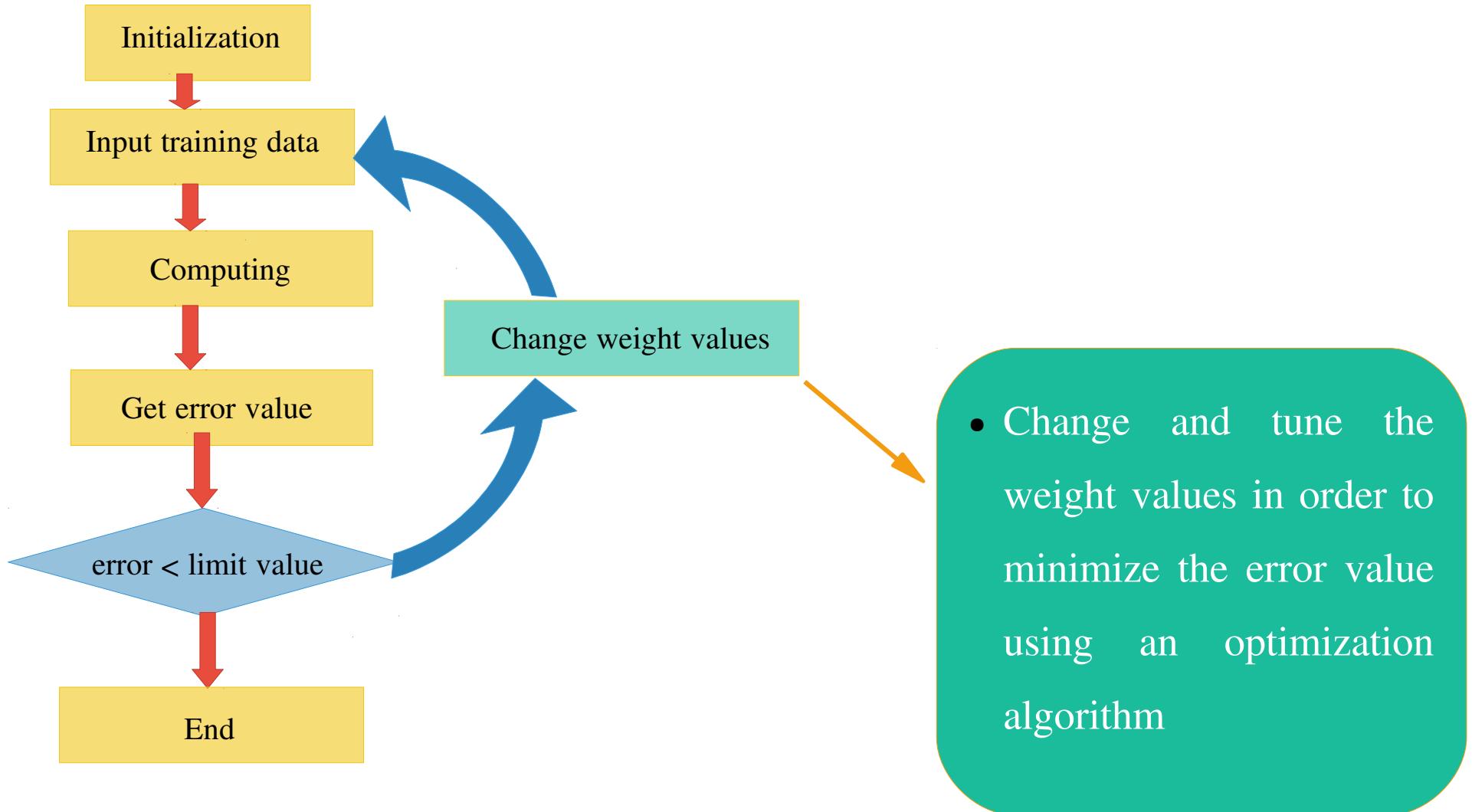


Training Flowchart

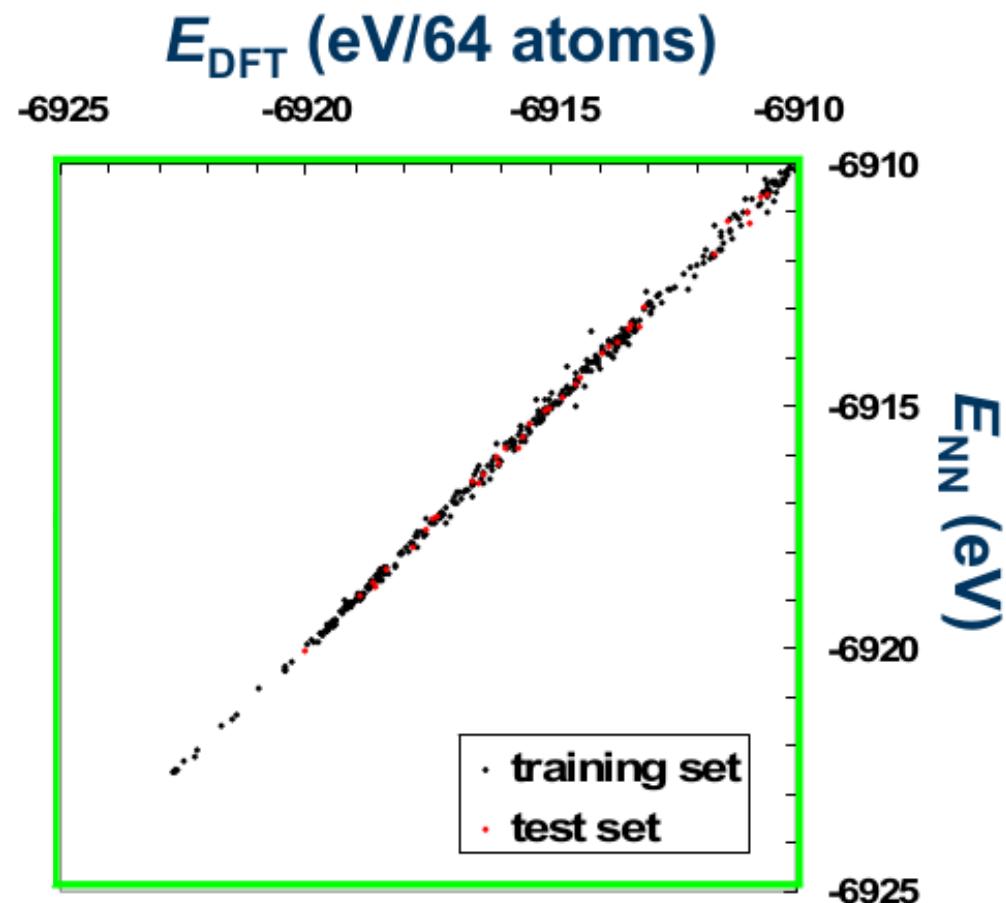


- Check the conditional statement: if error is enough small
YES => end
NO => change the weights using an optimization algorithm

Training Flowchart



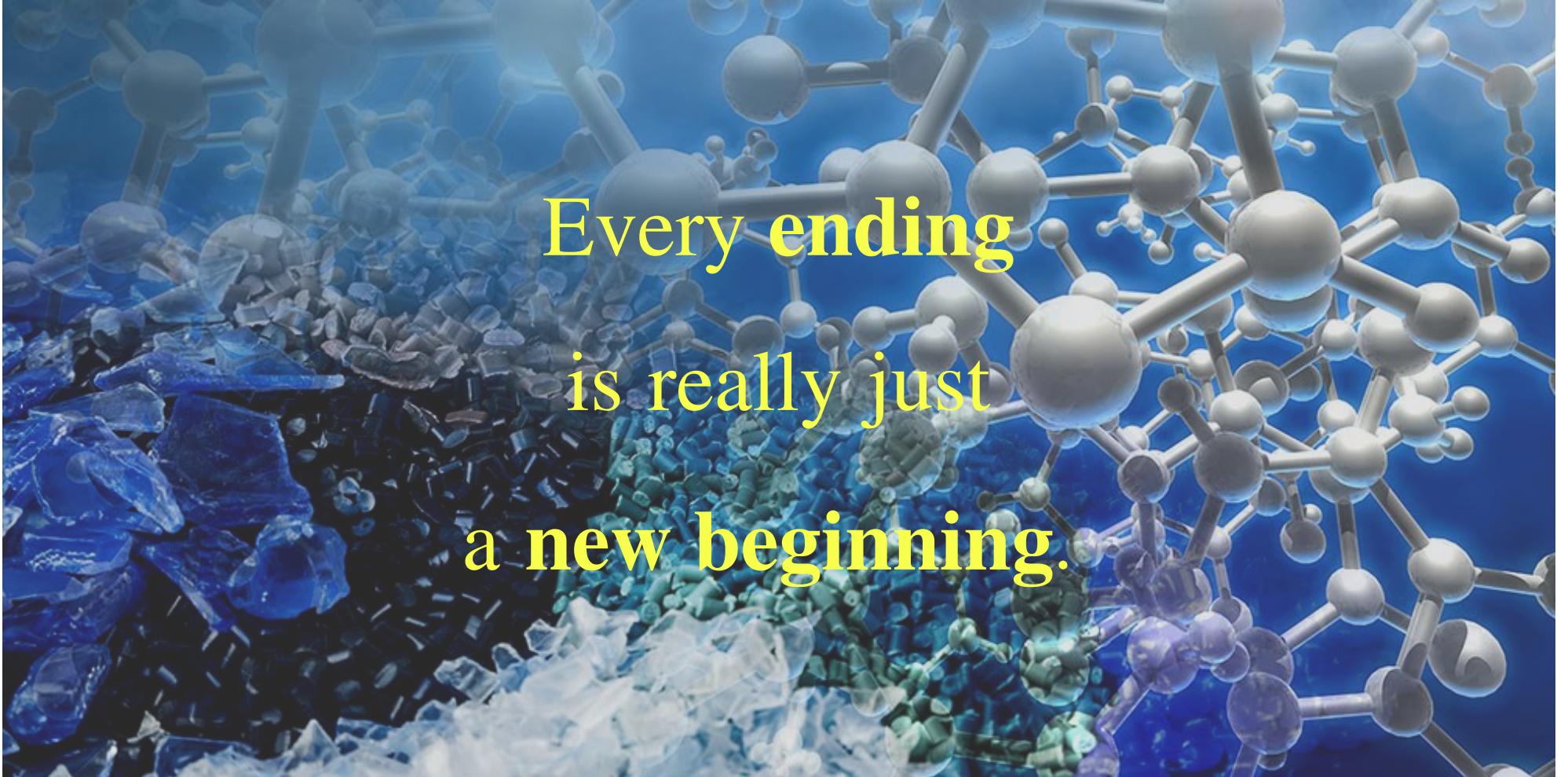
Best Training



Summary

- The problem of neural network learning can be seen as a function optimization problem, where we are trying to determine the best network parameters (weights and biases) in order to minimize network error.
- Several function optimization techniques can be directly applied to network learning, one of these techniques being the Levenberg-Marquardt algorithm.
- ANN is applicable to condensed systems (solids, large clusters, liquids) and reproduces total energies very accurately (also in value!)

Thank you!



Every ending
is really just
a new beginning.

13 Mehr 1397

13 Mehr 1397



جعفریان
نگرانی

دومین کارگاه
پردازی ماشینی در فیزیک:
Applications in Condensed Matter Physics
کاربردهای فیزیک ماده متموج

Generating PES for WATER molecule

Practical Session

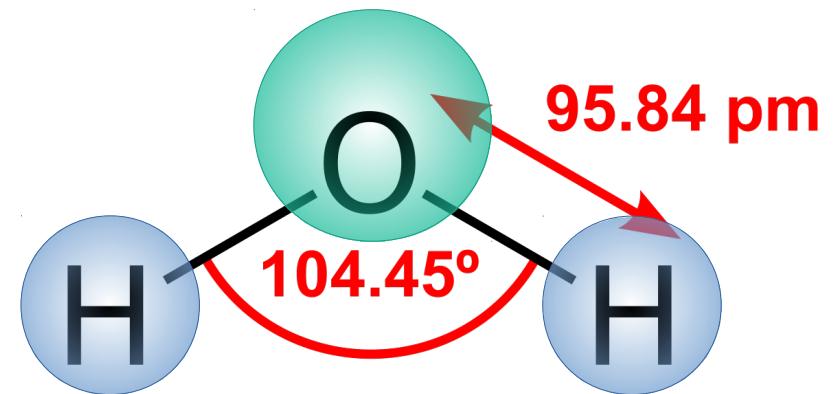
Samare Rostami

Robabe Rasoukhani

Project and Practice

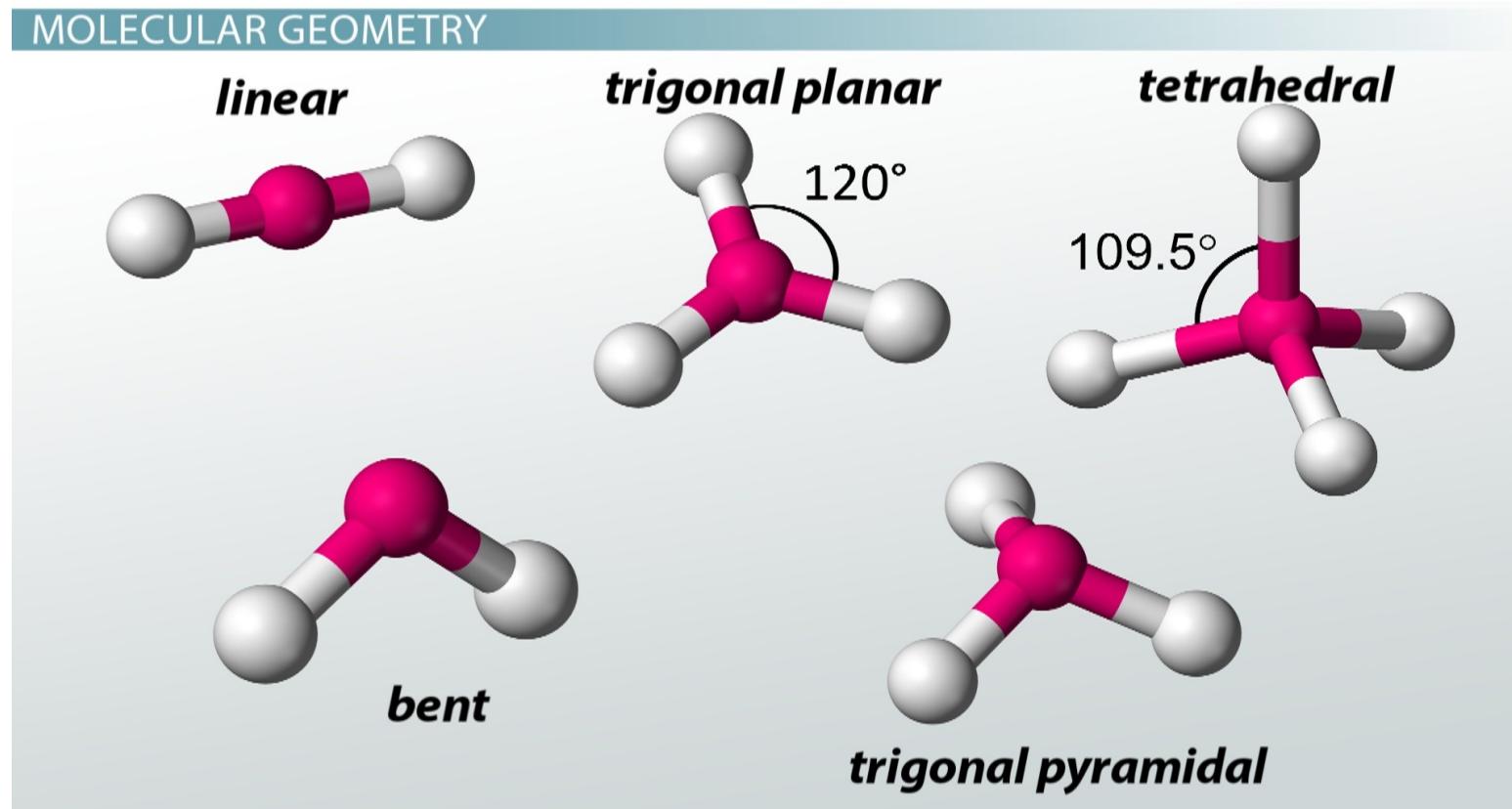
PROBLEM

- Generating PES for water (H_2O) molecule
- Finding the ground state of H_2O
- Finding the equilibrium bond length of H-O and bond angular between H-O-H of water molecule



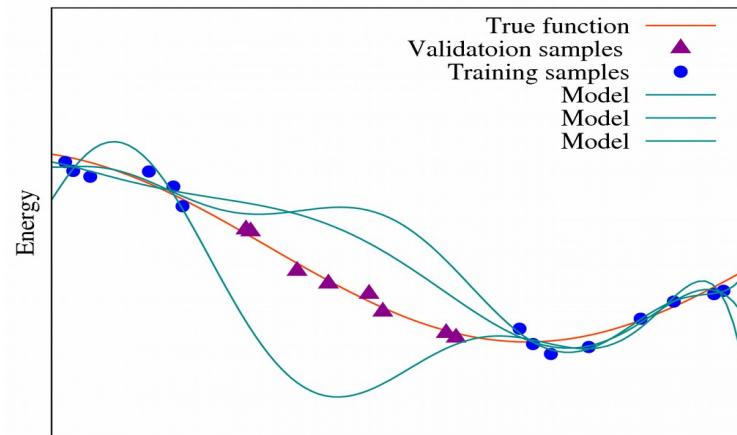
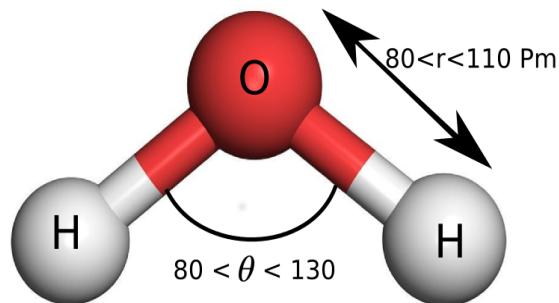
Preparation of data

1. Initial guess of molecular shape

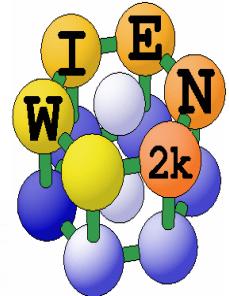


Preparation of data

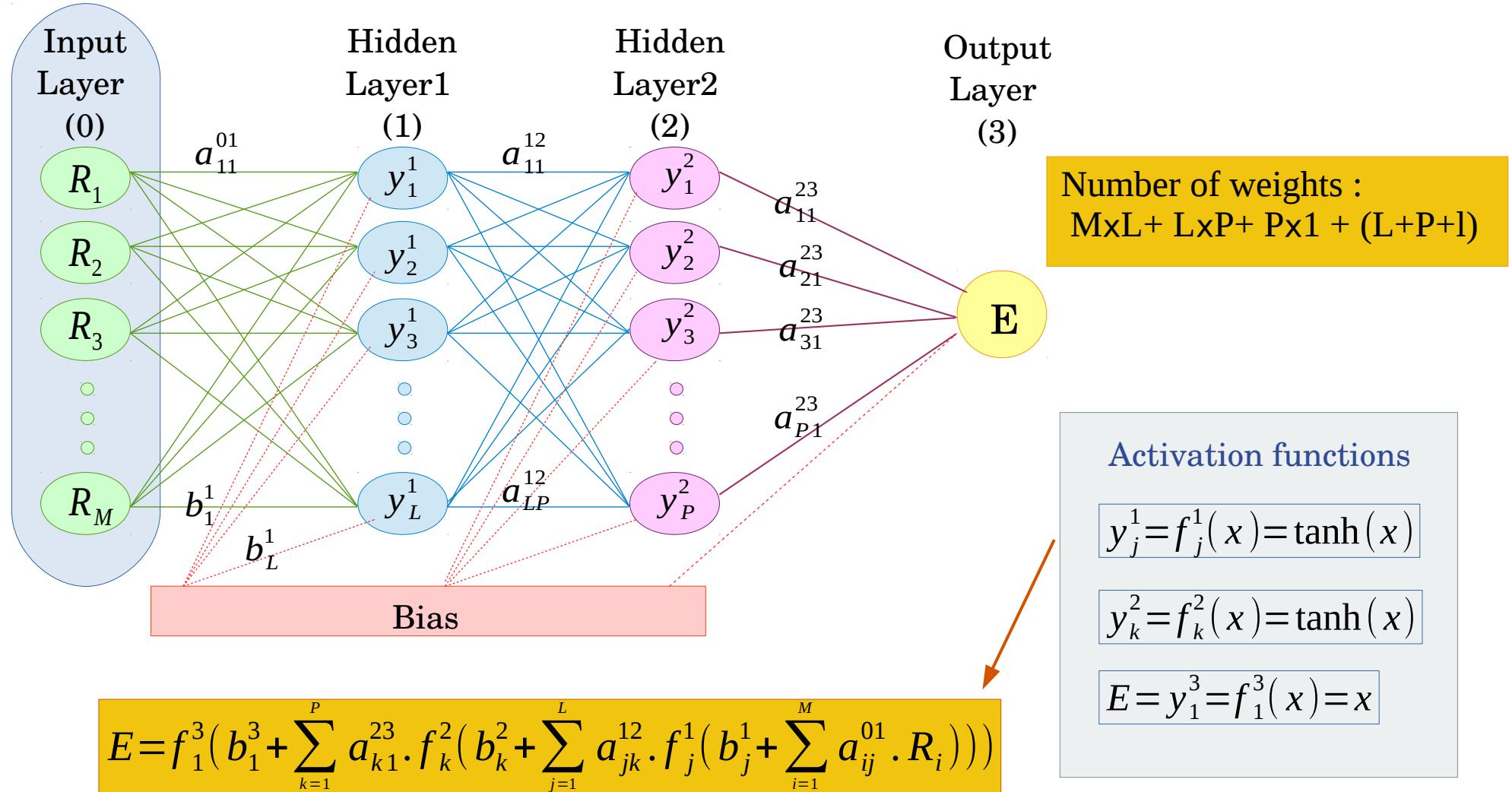
2. generation of diverse structures
close to molecular shape.



3. finding energies with a DFT package

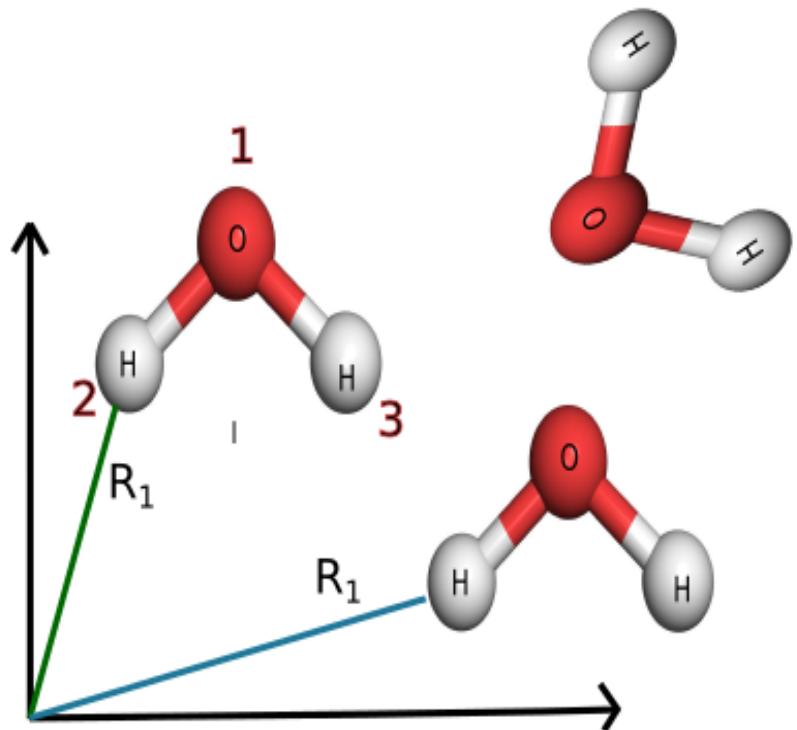


Structure of Neural Networks



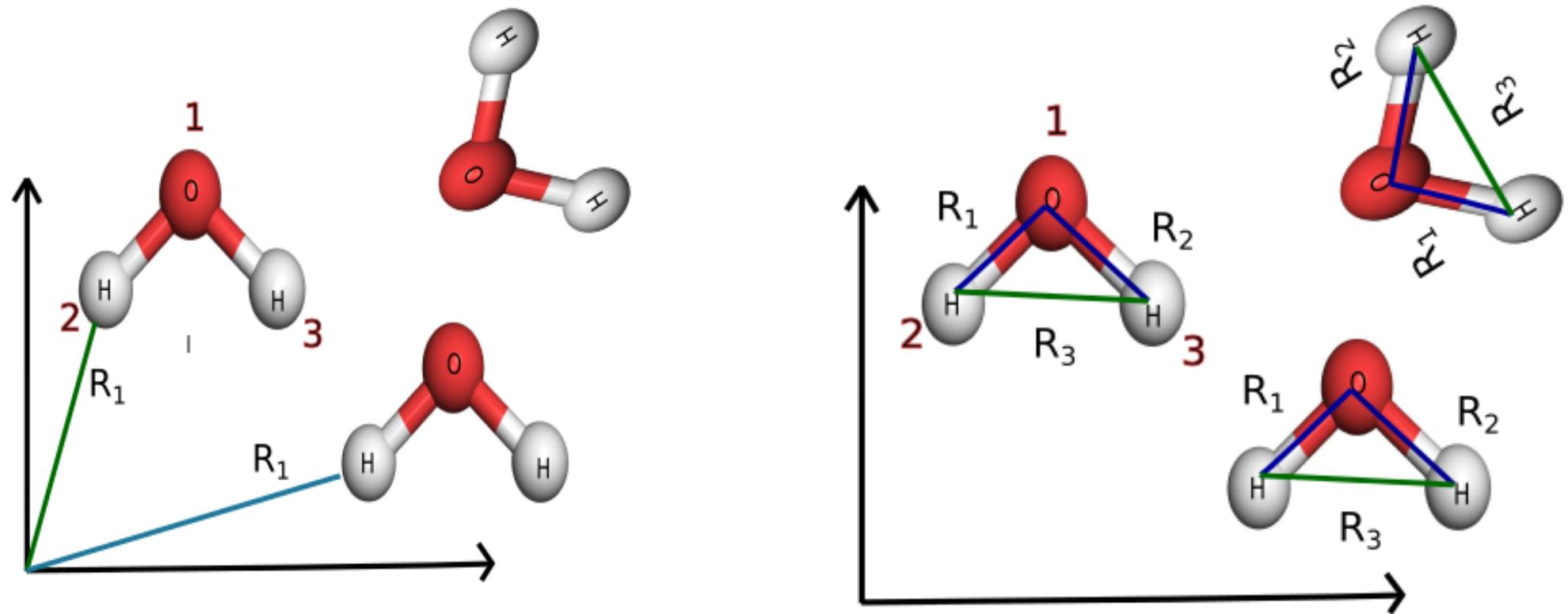
Input Layer of ANN

Input layer has to be invariant under rotation and translations

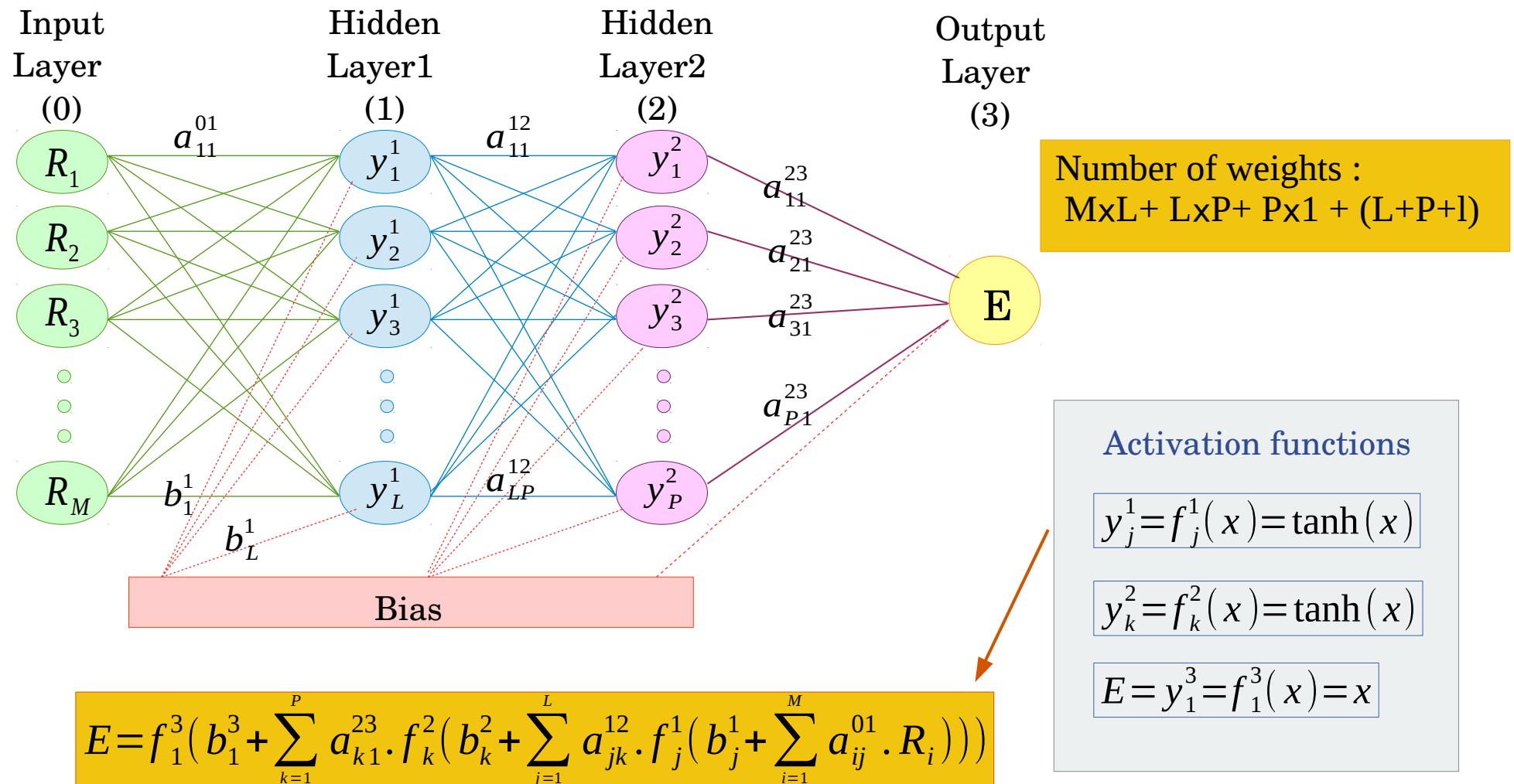


Input Layer of ANN

Input layer has to be invariant under rotation and translations



Structure of Neural Networks



```

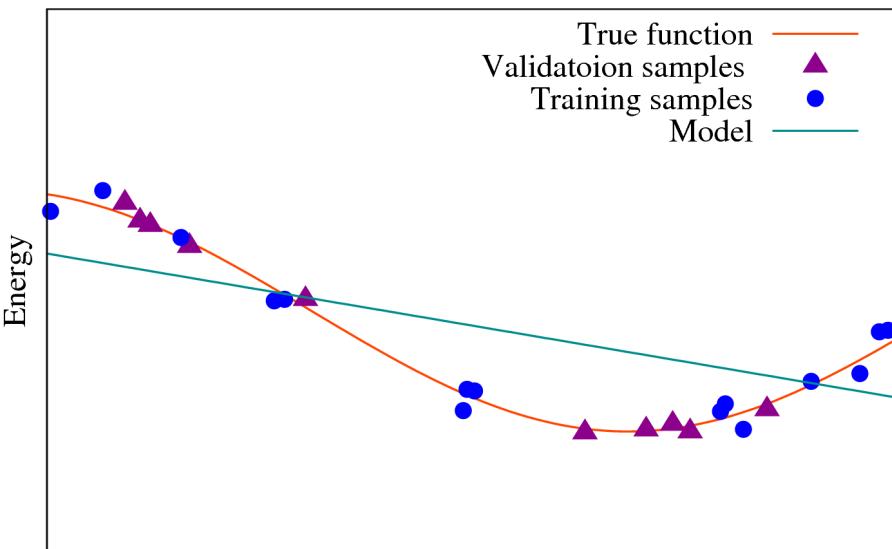
subroutine cal_architecture_2hiddenlayer(nl,nn,a,b,x,y,epot)
  implicit none
  integer, intent(in):: nl !number of hidden layer plus one
  integer, intent(in):: nn(0:10) !number of nodes in each hiddenlayer
  real(8), intent(inout):: a(140,140,10), b(140,10), x(140,10), y(140,0:10)
  real(8), intent(inout):: epot
  !local variables
  integer:: i, j, k
  real(8):: tt
  if(nl/=3) then
    write(*,'(a,i3)') 'ERROR: this routine works only for nl=3, while nl= ',nl
    stop
  endif
  !-----
  do j=1,nn(1) ! nn(1) = L
    tt=0.d0
    do i=1,nn(0) ! nn(0) = M : number of atoms
      tt=tt+a(i,j,1)*y(i,0)
    enddo
    x(j,1)=b(j,1)+tt
    y(j,1)=tanh(x(j,1))
  enddo
  !-----
  do k=1,nn(2) ! nn(2) = P
    tt=0.d0
    do j=1,nn(1)
      tt=tt+a(j,k,2)*y(j,1)
    enddo
    x(k,2)=b(k,2)+tt
    y(k,2)=tanh(x(k,2))
  enddo
  !-----
  tt=0.d0
  do k=1,nn(2)
    tt=tt+a(k,1,3)*y(k,2)
  enddo
  x(1,3)=b(1,3)+tt
  y(1,3)=x(1,3)
  !
  epot=y(1,3)

end subroutine cal_architecture_2hiddenlayer

```

4. Dividing configurations in two groups : training and validation

underfitting



overfitting

