TÜBİTAK

Turkish Journal of Physics

http://journals.tubitak.gov.tr/physics/

Research Article

Turk J Phys
(2016) 40: 231 – 243
© TÜBİTAK
doi:10.3906/fiz-1512-4

An approach based on neural computation to simulate transition metals using tight binding measurements

Adel BELAYADI[1,2,*], Boualem BOURAHLA[3], Leila AIT-GOUGAM[2], Fawzia MEKIDECHE-CHAFA[2]

[1]Laboratory of Coating Materials and Environment, LRME, Boumerdes, Algeria
[2]Department of Theoretical Physics, Faculty of Physics, University of Sciences and Technology Houari Boumediene, Algiers, Algeria
[3]Department of Quantum Chemistry, Faculty of Physics, M. Mammeri University, Tizi-Ouzou, Algeria

Received: 09.12.2015 • **Accepted/Published Online:** 15.03.2016 • **Final Version:** 01.12.2016

Abstract: A theoretical study of neural networks modeling, based on the tight binding approach, is proposed in this study. The aim of the present contribution is to establish a network topology to compute the binding energy parameters of transition metals. However, because of the different types of crystallization fcc, bcc, hcp, and sc of transition metals, neural network topology determination cannot be easily established, i.e. it would not be able to collect the data to feed the neurocomputing model. Hence, in order to overcome this problem, it would be helpful to distinguish one common structure from fcc, bcc, hcp, and sc. We observe that the structures fcc, bcc, and sc on (111) sheet and hcp on (0001) sheet form one common structure that is a two-dimensional hexagonal lattice. As the first application, the applicability of this choice of two-dimensional hexagonal lattice has been demonstrated by the ability to build a neurocomputing model able to determine the energy band structures of transition metals. Once the architecture is established, a second investigation will show the capability of data development techniques, by using the proposed common structure in our model, to calculate the binding energy parameters for transition metal atoms.

Key words: Neurocomputing, tight binding theory, transition metals, artificial neural networks

1. Introduction

Many fields of science applications make use of modeling techniques to analyze the behavior of complex systems from physical measurements [1–3]. This problem of physical modeling can be advantageously treated as a neurocomputing model. These new techniques have proven to be very suitable for solving such problems with fewer adjustable parameters [4,5]. They are considered an important class that has been attracting significant interest for many purposes. They are a suitable means for the problem of the analytical description of a set of data that appear in many sciences and applications and can be identified as a data-modeling problem or of a given system's identification. The desired task is generally achieved by feeding the neural models by several input–output values to expect one or more output quantity through a learning process that consists of adjusting the synaptic weights by using many learning algorithms to update these weights [6–8]. The algorithms of second order such as Levenberg–Marquardt [9], which take into account the second derivative, are ranked among the most efficient learning algorithms, because they give good modeling of the systems. In addition, the choice of the activation function is as important as the choice of the network architecture and the learning algorithms [10].

*Correspondence: adelphys@gmail.com

The sigmoid has shown its importance for the performance of a neuron network model [4] as the best activation function. Several studies have been applying neurocomputing models for predicting some parameters, for given systems [11–13]. Encouraged by these studies, we wanted to investigate an application of an artificial neural network to compute the physical parameters: $\epsilon_d, V_{dd\sigma}, V_{dd\pi}, V_{dd\delta}$, and r_d, which are the d-state energy, hopping energies, and the d-radius of d-orbital for transition metals. Additionally, as the process of these techniques requires collecting data to feed the neurocomputing model, we will use the tight binding approach to get our set of data, in order to start the training stage.

Tight binding theory has been applied to a wide variety of solids as an efficient, simple, and transparent model for the description of energy band structures [14–16]. The method is an approach to the calculation of the electronic band structure by using approximate wave functions based upon the linear combination of atomic orbitals for expanding the crystal wave functions [17,18]. The model also provides particular features of the energy bands, density of states, etc. The tight binding method used in this work is equivalent to that of the Slater and Koster framework [18]. This paper contains the famous Slater–Koster table that is used to build up the tight binding Hamiltonian that we will deal with in this study.

In this work, we will develop a theoretical study of a neurocomputing model for transition metals that involves the supervised learning technique by using second order algorithms to update the weights, and the sigmoid as an activation function. Furthermore, the implementation of the tight binding calculations will be used to feed in data to the proposed neurocomputing models. This paper is organized as follows: first, we apply the tight binding method, which is considered a simple way to obtain useful information about energy bands of transition metals with a two-dimensional hexagonal lattice. Second, in such an approach of neurocomputing an adequate description of metals under study will be proved by the possibility of building artificial neural networks that are able to optimize the training data given from tight binding calculations. Thirdly, application of an artificial neural network computes the physical parameters $\epsilon_d, V_{dd\sigma}, V_{dd\pi}, V_{dd\delta}$ and r_d, which are the d-state energy, hopping energies, and the d-radius of d-orbital, and then compares them with the well-known universal two center integrals given in Harrison tight binding theory [19]. Finally, a summary and a conclusion are provided.

2. Tight binding calculations to feed in data to artificial neural networks

A very high percentage of elements are metals. We will only focus on the transition metals, which are elements in the d-block of the periodic table [20]. The transition metals under study are Sc, Ti, V, Cr, Mn, Fe, Co, Ni, Cu, Zn, Y, Zr, Nb, Mo, Tc, Ru, Rh, Pd, Ag, Hf, Ta, W, Re, Os, Ir, Pt, and Au. These metals are mentioned and described by d-state in Harrison's framework [19]. The elements Ni, Cu, Rh, Pd, Ag, Ir, and Or cluster with fcc-type structures, whereas V, Cr, Fe, Nb, Mo, Ta, W, and Pt have bcc structures; however, Sc, Ti, Co, Zn, Y, Zr, Tc, Ru, Hf, Re, and Os cluster with hcp structures, and finally, Mn has the sc structure. In addition, we observe that the structures fcc, bcc, and sc on (111) sheet, and hcp on (0001) sheet form one common scheme, which is the two-dimensional hexagonal lattice, as shown in Figure 1. For the hcp sheet, it corresponds to a surface plane that intersects only the c-axis, being coplanar with the other 3 axes, i.e. it corresponds to the close packed planes of hexagonally arranged atoms that form the basis of the structure. For this case, we have the same pictures but the atoms are touching each other.

In the context of neural computing modeling, it can be useful to distinguish one common scheme from fcc, bcc, hcp, and sc that will be able to give an ability to build a neurocomputing model. This means the possibility of collecting a set of data that will be generated to feed the neural network model. Using different

structures cannot be easily modulated by the neural network methodology, i.e. the networks would be hopelessly overparameterized. For this reason and as a theoretical study, we will deal in this work with a two-dimensional hexagonal lattice.

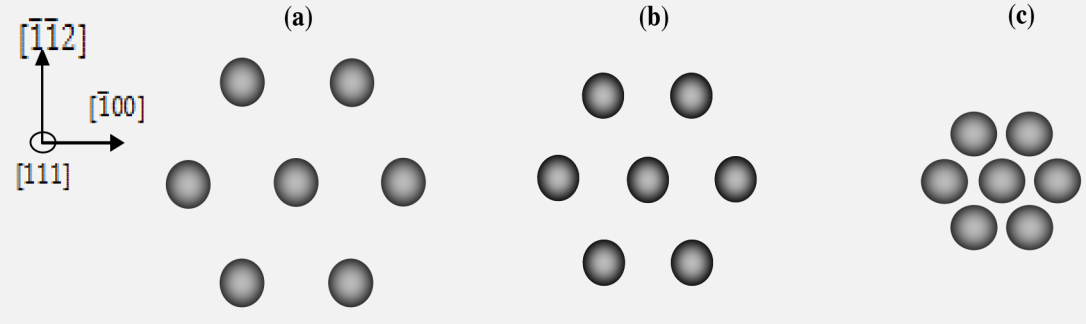

Figure 1. A schematic representation of two-dimensional hexagonal lattice given by cutting the (111) plane of: (a) The simple cubic (sc) lattice. (b) The body centered cubic (bcc) lattice. (c) The face centered cubic (fcc) lattice.

In this section, we implement the tight binding calculations for the mentioned metals with two-dimensional hexagonal lattice. The wave function is taken as linear combinations of atomic orbitals.

$$\psi(\mathbf{r},\mathbf{k}) = \sum_{i=1} C_i(\mathbf{k}) \left[\frac{1}{\sqrt{N}} \sum_{\mathbf{R}_n} e^{i\mathbf{k}.\mathbf{R}_n} \phi_n^i (\mathbf{r}-\mathbf{R}_i) \right], \tag{1}$$

where $\mathbf{R}_n$ are the space primitive unit cell vectors, n takes the values of the angular momentum character, i.e. denotes the type of atomic orbitals, C is a constant of normalization, $\mathbf{k}$ is the reciprocal lattice vector, and the ϕ_n^i functions are chosen to have the same symmetry properties as the atomic orbitals. Since our unit cell contains only one atom, the unit cell vectors are at the same time the nearest neighbor vectors, as shown in Figure 2.

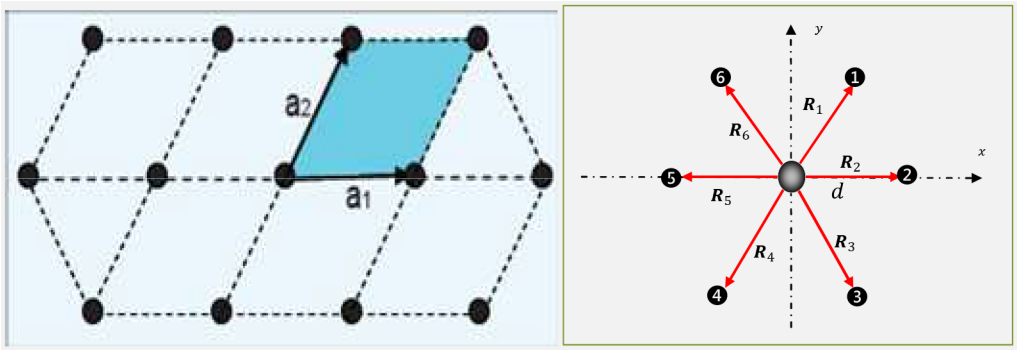

Figure 2. Left: Schematic representation of the periodic lattice, which contains one atom per unit cell. Right: The hexagonal atomic plane denoting the direct lattice vectors $\mathbf{R}_n$ between the first nearest neighbors.

$$\mathbf{a}_1 = d\hat{x}, \mathbf{a}_2 = d\left(\frac{1}{2}\hat{x} + \frac{\sqrt{3}}{2}\hat{y}\right)$$

$$\mathbf{R}_1{=}d\left(\frac{1}{2},\frac{\sqrt{3}}{2}\right), \mathbf{R}_2{=}d\,(1,0)\,\mathbf{R}_3{=}d\left(1/2,-\sqrt{3}/2\right), \mathbf{R}_4{=}d\left(-1/2,-\sqrt{3}/2\right)$$

$$\mathbf{R}_5{=}d\left(-1,0\right),\mathbf{R}_6{=}d\left(-1/2,\sqrt{3}/2\right), \tag{2}$$

where d represents the hexagonal lattice spacing.

We implement our study with one atom per cell and five orbitals per atom, representing the five atomic orbitals $d_{xy}, d_{xz}, d_{yz}, d_{x^2-y^2}$, and d_{z^2}. The (5×5) secular determinants representing all possible nearest neighbor interactions between d-states are given as

$$\sum_m H_{mn}\left(\mathbf{k}\right)C_n\left(\mathbf{k}\right) = E\sum_m \delta_{mn}C_n\left(\mathbf{k}\right), \tag{3}$$

where H_{mn} are integrals derived by making use of the Slater–Koster table that are iteratomic matrix elements as a function of direction cosines [18].

Finally, the solution of the eigenvalue problem in Eq. (3) gives rise to five energy bands, $E_1(\mathbf{k}), E_2(\mathbf{k}), E_3(\mathbf{k}), E_4(\mathbf{k})$, and $E_5(\mathbf{k})$, along the wave vector $\mathbf{k}$ in the reciprocal space lattice. These energy bands will be used as inputs to generate the neurocomputing models. The descriptions of generating these energy values are given in the next sections.

3. Neural network models

The neural network models are inspired by the neural physiology of the human brain by combining the universality of associative thinking with the precision of a mathematical model [21]. Figure 3 shows a schematic representation of a feed forward neural network with q input parameters, an immediate layer of n nodes, and p output parameters. The connections are from each input node to each intermediate layer node and from each intermediate layer node to each output node in a feed-forward manner.

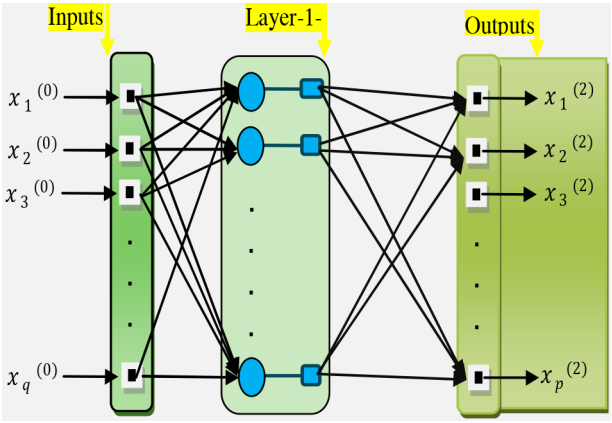

Figure 3. A schematic representation of one hidden layer neural network, where $x_q^{(0)}$ and $x_p^{(2)}$ represent, respectively, the inputs and outputs of the neural model.

Mathematically, the output of the ith neuron in the kth layer of a network is [22]

$$x_i^{(k)} = f^{(k)}\left(b^{(k)} + \sum_{j=1}^{\tau} w_{ij}^{(k)} x_j^{(k-1)}\right), \tag{4}$$

where b is a bias term, and w_{ij} are weight coefficients that are connection links between the nodes in different layers, and τ is the number of neurons in the layer $(k-1)$.

The node of each neuron in the network is associated with a continuously differentiable transfer function that modulates the sum of the weighted inputs. Different types of transfer functions can be employed in different layers. In this study, the sigmoid function has been employed in hidden layers, i.e. $f(u) = 1/(1-e^{-u})$ However, for the output layer a linear function has been used for activation, i.e. $f(u) = u$

3.1. Database generation and training algorithm

In this step, we will give a brief description of the collection of the parameters that will be used for the neural network database generation. The five energy bands data, which are collected from the solution of Eq. (3) along the $\Gamma - K$ path, will be considered a first collection that depends on the various variables given as follows: $V_{dd\sigma}, V_{dd\pi}, V_{dd\delta}, \epsilon_d$, and r_d quantities.

These five variables will be considered a second collection. The details of data selection are given in section 3.2.

In the term of d-orbitals, the Harrison universal two-center integrals that were used to solve the eigenvalue problem have the following expressions [18,19]:

$$V_{dd\sigma} = -\frac{45}{\pi}\frac{\hbar^2 r_d^3}{md^5}, V_{dd\pi} = \frac{30}{\pi}\frac{\hbar^2 r_d^3}{md^5} V_{dd\delta} = -\frac{15}{2\pi}\frac{\hbar^2 r_d^3}{md^5}, \tag{5}$$

where m denotes the electron vacuum mass, d depicts the nearest-neighbor iteratomic distance, and r_d is the d-radius.

A brief summary of a part of the database is presented in Table 1; this part of the data is explicitly given in refs [19,20].

3.2. Normalization processes of the database

The process of normalizing transforms the user input data to a form that is easier or more efficient for a network. This process of standardization is a common practice in the application of artificial neural networks, and it was done by transforming the data to fall in the interval $x_i \in [-1, 1]$, by using the following expression:

$$x'_i = (x_i - \bar{x}_i)/\sigma_i, \tag{6}$$

where x'_i are the normalized values of the ith input parameters, and x_i are the actual values of the parameters before the application of normalization processes, which make use of the mean and standard deviations $\bar{x}_i$ and σ_i.

3.3. Training processes of second-order algorithm, Levenberg–Marquardt

Neural network performance is reached by the correctness of the weight values using the Levenberg–Marquardt algorithm. The theoretical formalism of this weight update rule is given as follows [23,24].

Considering a neuron that receives q input parameters connected by a weight vector $\mathbf{w}^T \in \mathbf{R}^{q\times 1}$. The update during the kth iteration would be as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mathbf{H}^{-1}(k)\mathbf{g}(k). \tag{7}$$

Table 1. Parameters of transition metals under study that were used for the neural network database generation; the d-radius (r_d) values are tabulated from the atomic-surface method (ASM) from Straub and Harrison (1985), ϵ_d represents the d-state energy given from the free-electron measurement. The values of the latter parameters are from Harrison's book [19], and the values of universal two-center interactions $V_{dd\sigma}, V_{dd\pi}, V_{dd\delta}$ are given from Eq. (5).

Name and structure	Electronic configuration	Atomic distance	$r_d(\dot{A})$	$\epsilon_d(eV)$	$V_{dd\sigma}(eV)$	$V_{dd\pi}(eV)$	$V_{dd\delta}(eV)$
Scandium (Sc) hsp	$[Ar]\,3d^1 4s^2$	3.310	1.163	–9.35	–2.444	1.629	–0.407
Titanium (Ti) hsp	$[Ar]\,3d^2 4s^2$	2.950	1.029	–11.05	–3.011	2.007	–0.501
Vanadium (V) bcc	$[Ar]\,3d^3 4s^2$	3.020	0.934	–12.54	–0.726	0.484	–0.121
Chromium (Cr) bcc	$[Ar]\,3d^5 4s^1$	2.880	0.939	–13.88	–0.936	0.624	–0.156
Manganese (Mn) cs	$[Ar]\,3d^5 4s^2$	8.890	0.799	–15.27	–0.001	0.0006	–0.0001
Iron (Fe) bcc	$[Ar]\,3d^6 4s^2$	2.870	0.744	–16.54	–0.473	0.315	–0.079
Cobalt (Co) hcp	$[Ar]\,3d^7 4s^2$	2.510	0.696	–17.77	–2.089	1.393	–0.348
Nickel (Ni) fcc	$[Ar]\,3d^8 4s^2$	3.520	0.652	–18.97	–0.316	0.211	–0.052
Copper (Cu) fcc	$[Ar]\,3d^{10} 4s^1$	3.610	0.688	–20.26	–0.638	0.425	–0.106
Zinc (Zn) hsp	$[Ar]\,3d^{10} 4s^2$	2.660	1.415	–8.46	–3.135	8.757	–2.189
Yttrium (Y) hcp	$[Kr]\,4d^1 5s^2$	3.650	1.602	–6.80	–3.918	2.612	–0.653
Zirconium (Zr) hcp	$[Kr]\,4d^2 5s^2$	3.230	1.415	–8.46	–4.975	3.317	–0.829
Niobium (Nb) bcc	$[Kr]\,4d^4 5s^1$	3.300	1.328	–9.980	–1.340	0.893	–0.223
Molybdenum (Mo) bcc	$[Kr]\,4d^5 5s^1$	3.150	1.231	–11.49	–1.347	0.898	–0.224
Technetium (Tc) hcp	$[Kr]\,4d^5 5s^2$	2.740	1.109	–13.08	–5.453	3.635	–0.908
Ruthenium (Ru) hcp	$[Kr]\,4d^7 5s^1$	2.700	1.083	–14.61	–5.465	3.643	–0.911
Rhodium (Rh) fcc	$[Kr]\,4d^8 5s^1$	3.800	1.020	–16.16	–0.826	0.551	–0.137
Palladium (Pd) fcc	$[Kr]\,4d^{10}$	3.890	1.008	–17.70	–0.709	0.473	–0.118
Silver (Ag) Fcc	$[Kr]\,4d^{10} 5s^1$	4.090	0.889	–19.23	–0.379	0.252	–0.063
Hafnium (Hf) bcc	$[Kr]\,4f^{14} 5d^2 6s^2$	3.200	1.455	–8.14	–5.668	3.778	–0.944
Tantalum (Ta) bcc	$[Kr]\,4f^{14} 5d^3 6s^2$	3.310	1.346	–9.57	–1.375	0.916	–0.229
Tungsten (W) bcc	$[Kr]\,4f^{14} 5d^4 6s^2$	3.160	1.268	–10.97	–1.449	0.966	–0.241
Rhenium (Re) hcp	$[Kr]\,4f^{14} 5d^5 6s^2$	2.760	1.201	–12.35	–6.678	4.452	–1.113

The Hessian matrix $\mathbf{H} \in \mathbf{R}^{q\times q}$ can be approximated as $\mathbf{H} = \mathbf{J}^T\mathbf{J}$, and the gradient can be computed as $\mathbf{g} = \mathbf{J}^T\mathbf{e}$, where $\mathbf{J}$ is the Jacodian matrix, which contains first derivatives of the network errors, and $\mathbf{e}$ is a vector of mean square network errors. The Jacobian matrix can be computed through a standard back propagation technique that is less complex than computing the Hessian matrix. By using this approximation, we obtain the Levenberg–Marquardt update as

$$\mathbf{w}\,(k+1) = \mathbf{w}\,(k) - \left[\mathbf{J}^T\,(k)\,\mathbf{J}\,(k) + \mu\mathbf{I}\right]\mathbf{J}^T\,(k)\,\mathbf{e}(k), \tag{8}$$

where μ is the learning parameter, and $\mathbf{I} \in \mathbf{R}^{q\times q}$ is the identity matrix.

As a final step, when k tends to infinity, $\mathbf{e}\,(k)$ tends to zero, and then $\mathbf{w}\,(k)$tends to the right weight matrix of network that permits modulation of the outputs accurately. The training and evaluation of a neurocomputing model involves three phases: firstly, collecting the database generations, which characterize the system to be modulated; secondly, construction of the proposed network topology for the systems under study; thirdly, the network training and testing operations will be checked. The essentials of these steps for determination of weights are discussed later.

4. Artificial neural networks topology and network training

4.1. Network testing and validation of modulating the two-hexagonal lattice

The work in this stage describes the methodology to extract the band energies of two-hexagonal lattice for transition metals by using neural network models. The choice of a particular network topology depends on the complexity of the problem. The number of network layers and the number of neurons in each layer determine the ability of the network to capture the trends in the data. A schematic representation of the network topology is shown in Figure 4.

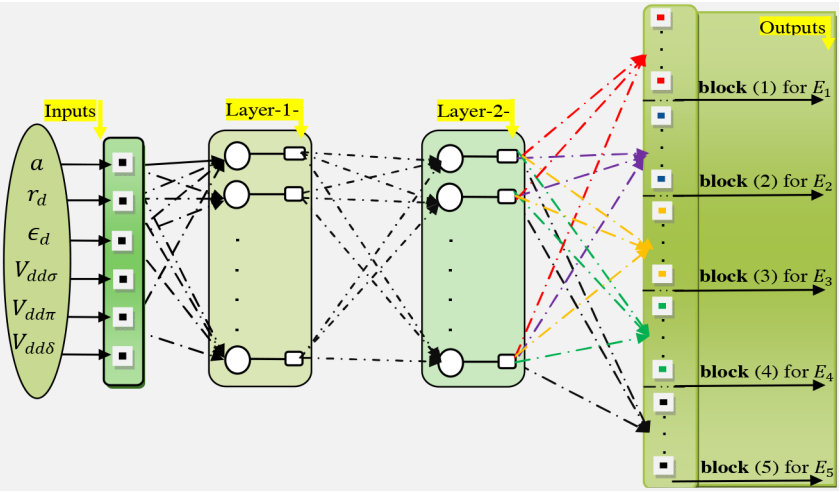

Figure 4. A schematic representation of two hidden layers of an artificial neural network.

The important factor that influences the learning process of the network is the quality of the learning set (examples that constitute the training data) and the diversity of values. Indeed, the neural network will have a good performance (is more likely to respond correctly) if the sampling quality is better and the data examples of learning are diverse. In this approach we selected our data set as follows:

- The output layers contain the values of the five energy vectors $E_1(\mathbf{k}), E_2(\mathbf{k}), E_3(\mathbf{k}), E_4(\mathbf{k})$, and $E_5(\mathbf{k})$ generated from the tight binding approach. These energy values are organized in five blocks in order to be used as training data.

- Each block contains the values of one energy band. For instance, the block (i) will take the values of the energy vector $E_i(\mathbf{k})$ along the wave vector $\mathbf{k}$ in the reciprocal space lattice.

- The wave vector $\mathbf{k}$ in the reciprocal space lattice is taken along the high symmetry points $\Gamma - K$. Then we sample each energy vector $E_i(\mathbf{k})$ to yield 200 training data generated uniformly over the path $\Gamma - K$.

- The number of neurons in each block is limited to ten units, i.e. 10 values from each energy vector are generated uniformly over the path $\Gamma - K$ in such a way that during the training process 75% of the sampled values of the energy vector along the path $\Gamma - K$ will be included.

- Thus, as per this topology, the Harrison universal two-center interactions $(V_{dd\sigma}, V_{dd\pi}, V_{dd\delta})$ and the parameters d-radius r_d, d-state energy ϵ_d, and the interatomic distance a are used as input parameters.

- Additionally, the number of neurons in the first layer has been treated as a variable parameter, in order to find the optimal number of neurons that is adequate to ensure a higher accuracy of the model without overfitting the data or be stuck in local optima [7].

The performance and the results of the first network topology are shown in Figures 5 and 6.

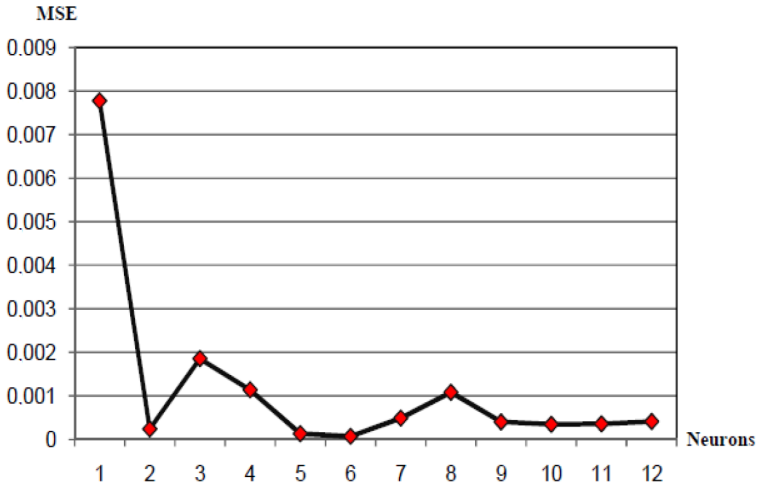

Figure 5. Plot of the mean square norm of the error MSE versus the number of neurons in the case of twelve neurons in the second hidden layer.

The results of the first network topology can be explained as follows:

⋆ Figure 5 shows the mean square error (MSE) as a function of the number of neurons in the first hidden layer. Several network topologies were studied in which the number of neurons varied from 1 to 12. The obtained results indicate that an increase in the number of neurons would lead to a decrease in the MSE; however, the best architecture to capture our data was obtained through using six neurons in the first hidden layer and twelve neurons in the second hidden layer, where the MSE tends to $\times 610^{-4}$.

⋆ The performance of the neural network for the training data is given by using only the topology in which the number of neurons was powered by six in the first hidden layer "optimal case". Furthermore, the initialization of the network weights and bias were generated by using Nguyen and Widrow's initialization algorithm [25]. The quality of the network calculation (the network response and the desired response) is shown in Figure 6.

⋆ As we said before, the modeling by the neural network requires collecting the training input–output units (values of the given inputs and responses) or the networks will be hopelessly overparameterized. In order to overcome this problem of neural computing modeling, the proposed common scheme, which is a two-dimensional hexagonal lattice, has given good agreement to modulate the listed metals, because it allows collecting enough information to feed in data to the input–output units of the neural network. As seen in Figure 6, the values of the training output quantities given through the neural model are quite fair compared to the desired tight binding calculations, hence the ability to build up a neurocomputing model for our transition metals by using the two-dimensional hexagonal lattice.

⋆ According to the performance of the proposed neural computing model, we will deal in the next section with the two-dimensional hexagonal lattice, and, as a theoretical study, the following development and

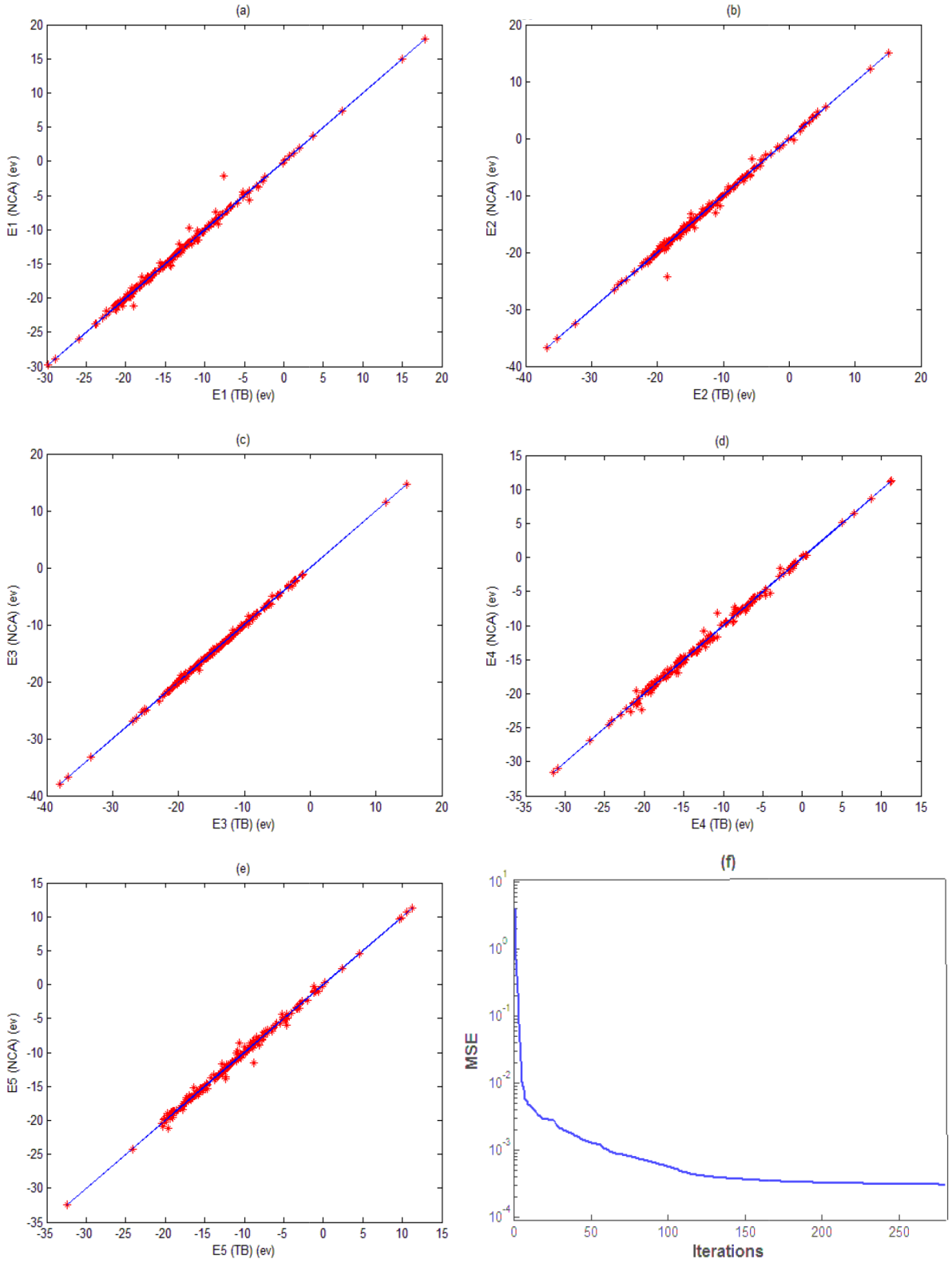

Figure 6. (a–e) The tight-binding TB calculations: $E_1(TB)E_2(TB)E_3(TB)E_4(TB)$, and $E_5(TB)$ versus the neuro-computing application (NCA) calculations: $E_1\,(NCA)$,$E_2(NCA)E_3(NCA)E_4(NCA)$, and $E_5(NCA)$. The depicted data are for the training set given in Table 1. (f) Plot of the mean square error versus the number of iterations.

evaluation of an artificial neural network will focus on the capability of development of data techniques to compute the quantities of interest: $V_{dd\sigma}, V_{dd\pi}, V_{dd\delta} \epsilon_d$, and d-radius r_d.

4.2. Neurocomputing model of the quantities $\mathbf{V_{dd\sigma}}$,$\mathbf{V_{dd\pi}}$,$\mathbf{V_{dd\delta}}$,$\mathbf{\epsilon_d}$, and $\mathbf{r_d}$

In this case, we will describe a particular network topology given in Figure 7.

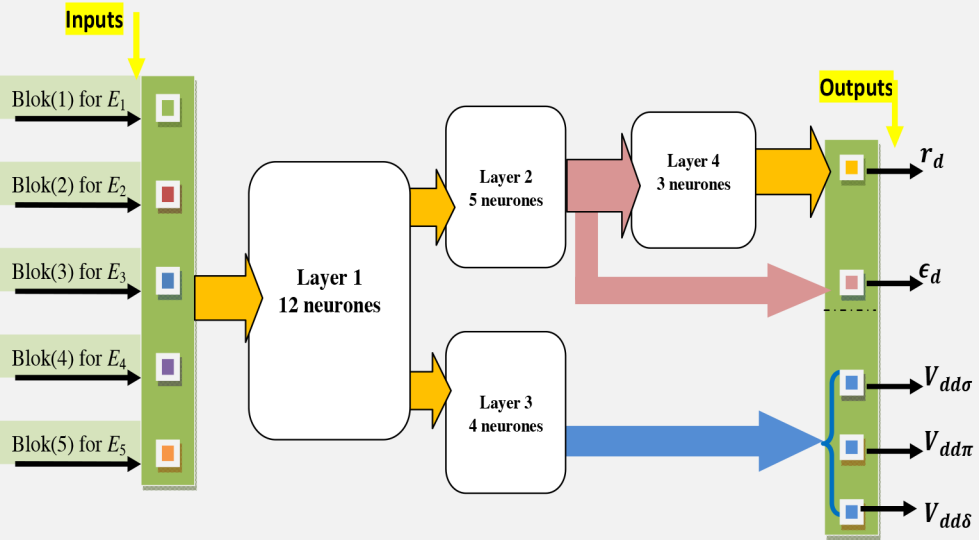

Figure 7. A schematic representation of three hidden layers of artificial neural network.

The inputs of this model will be generated by the output values given from the result of the first architectures in Figure 4 and the outputs will be fed with the help of Table 1. In this model, the number of neurons in the first layer was powered with twelve neurons, the second layer with five neurons, the third layer with four neurons, and the fourth layer with three neurons. The purpose of the present second architecture is to modulate the quantities $\epsilon_d, V_{dd\sigma}, V_{dd\pi}, V_{dd\delta}$, and r_d, which are the d-state energy, hopping energies, and the d-radius of d-orbital, and then comparing them with the universal two center integrals given in Harrison tight binding theory.

5. Results and discussion

For a better validation of our programs and in order to see if the results of the neural network model are reliable or not, the performance of the network model in estimating the quantities $V_{dd\sigma}, V_{dd\pi}, V_{dd\delta}$ and ϵ_d for the metals osmium (Os), iridium (Ir), platinum (Pt), and gold (Au) is given in Tables 2 and 3 as the testing step.

Additionally, to establish the credibility of the neural network approach, the d-state radius r_d given by the NCA approach was compared with d-state radius r_d calculation from muffin-tin-orbital (MTO) theory, and the tabulated d-state radius r_d from atomic-surface method (ASM). Figure 8 summarizes the three calculation approaches for all the elements under study.

The determination of on-site energy and hopping energy parameters is an important and crucial step to be taken, in order to be able to calculate the energy bands of metals either in the bulk or on the surface. The method given in Harrison's framework has been used as a theoretical approach to calculate the quantities $V_{dd\sigma}, V_{dd\pi}, V_{dd\delta}$, and ϵ_d. In fact, these transition metals parameters are well known; in contrast, we wanted in this approach to show the power of machine learning to predict these quantities and to explain how to select a

Table 2. Comparison of transition metals parameters of the quantities of interest $V_{dd\sigma}, V_{dd\pi} V_{dd\delta}$, and ϵ_d given by the neurocomputing application (NCA), and the values given from Harrison's tight-binding theory (HTB) from Eq. (5).

Parameters	Methods	Osmium (Os)	Iridium (Ir)	Platinum (Pt)	Gold (Au)
$\epsilon_d(eV)$	HTB	–13.7400	–15.1200	–16.4700	–17.7800
	NCA	–13.5369	–14.9388	–16.2966	–17.5541
$V_{dd\sigma}(eV)$	HTB	–5.9544	–0.9446	–0.2957	–0.5577
	NCA	–5.9088	–0.9308	–0.2917	–0.5826
$V_{dd\pi}(eV)$	HTB	3.9696	0.6297	0.1971	0.3718
	NCA	4.0009	0.6670	0.2435	0.4470
$V_{dd\delta}(eV)$	HTB	–0.9924	–0.1574	–0.0493	–0.0929
	NCA	–1.0009	–0.1607	–0.0362	–0.0850

Table 3. The first d-state radius (r_d) values obtained from muffin-tin-orbital (MTO) theory from Andersen and Jepsen (1977); the second d-radius (r_d) values are tabulated from the atomic-surface method (ASM) from Straub and Harrison (1985), the third d-radius (r_d) values are calculated by the neurocomputing model (NCA).

Parameters	Methods	Osmium (Os)	Iridium (Ir)	Platinum (Pt)	Gold (Au)
$r_d(\dot{A})$	MTO	1.219	1.159	1.116	1.081
	ASM	1.142	1.085	1.069	1.007
	NCA	1.1209	1.0652	1.0582	0.9923

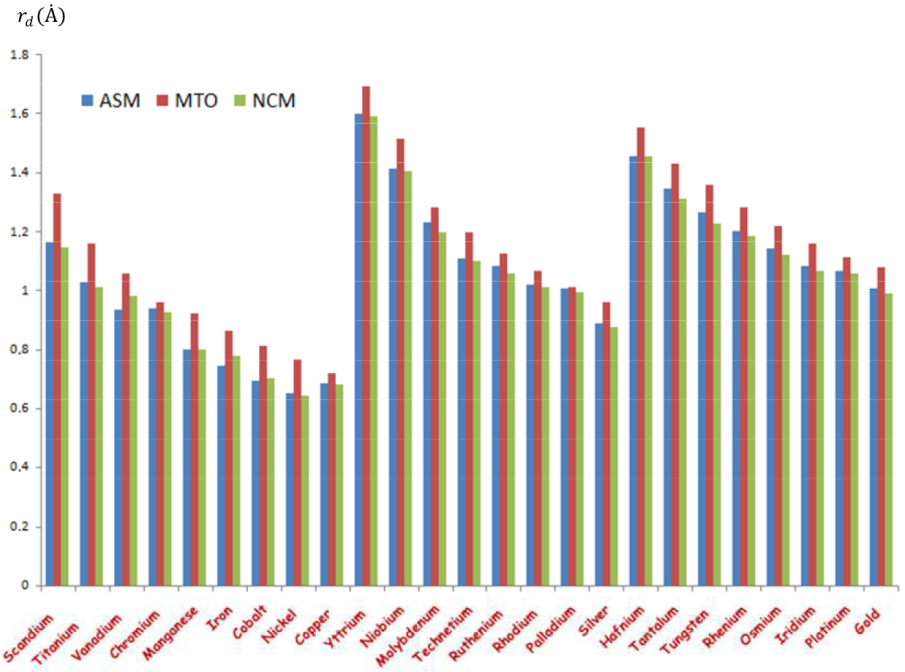

Figure 8. The d-state radius r_d values with red color are obtained from muffin-tin-orbital (MTO) theory from Andersen and Jepsen (1977); the d-radius (r_d) values in blue are tabulated from the atomic-surface method (ASM) from Straub and Harrison (1985) [19], and the d-radius (r_d) values in green are calculated by the neurocomputing model (NCA).

common scheme from the different structures fcc, bcc, and sc to be able to deal with the machinery of neural computing. It has been shown that the important advantage of choosing the two-hexagonal lattice as a common scheme is to collect the training data, which is a crucial step to build up our neural network machinery. The performance of the neural network model can be explained as follows:

- In order to get an optimal model, we can change the structure of the neural network in the hidden layers, yet things are much less easier. However, there is no law, no rules, and no theorem that would determine the number of neurons to place in the hidden layer for an optimal neural network. Several network topologies were proposed in which the number of neurons was varied in the first, second, and third layer. We have found that using ten neurons in the first hidden layer, twelve neurons in the second hidden layer, and six in the third layer provides an adequate architecture to capture our data.

- In our proposal methodology, the structure of our neural model was chosen in such a way that we have raised the neurons unit in the input layer (we have taken ten-neuron unit in each block as we have mentioned before) which allowed us to have more connections in terms of weights and bias. This representation is a reflection of the biological inspiration. In fact, the weight assigned to each unit measures these relative importance connections. This presentation has shown that neural network machinery is a pragmatic model that would permit improving interesting tasks in terms of accuracy and computation time.

- In the network learning processes, error back propagation (EBP) is considered one of the most used training algorithms for feed forward artificial neural networks. However, this algorithm is very slow if the size of the network is too large. Second order algorithms help to converge much faster than first order algorithms. Furthermore, by combining the training speed of second order algorithms, namely the Levenberg–Marquardt, and the stability of EBP algorithm, the mean square error reached 0.01% after the end of the training process. In addition, we obtained the right weight matrices of networks that permit the modulation of the quantities of interest.

For evaluating the performance of the neurocomputing model, the remaining metals, osmium (Os), iridium (Ir), platinum (Pt), and gold (Au), were evaluated. The performance of the neural network for the training and the test data is shown in Tables 2 and 3. It is obvious that all the output quantities of interest, $V_{dd\sigma}, V_{dd\pi}, V_{dd\delta}\epsilon_d$, and d-radius r_d, are in excellent agreement with the values given from Harrison's tight-binding theory.

6. Summary and conclusion

To sum up, we reported a theoretical study based of 27 elements of transition metals, clustered in different structures, fcc, bcc, hcp, and sc. The present paper has shown the ability of artificial neural networks theory to estimate the quantities of interest, $V_{dd\sigma}, V_{dd\pi}, V_{dd\delta}\epsilon_d$, and d-radius r_d, of transition metals by using tight binding measurements of energy bands.

Neural network modeling requires collecting the training input–output units. The proposed two-dimensional hexagonal lattice has given good agreement to modulate the cited metals, because it allows collecting enough information to feed in data to the input–output units of a neural network. Additionally, in our proposal methodology, we have also changed the structure of neural models by raising the neurons unit in the output layer, which permitted us to have more connections in terms of weights and bias. The idea was presented as generating the input–output units (the number of neurons in each block is limited to ten units, i.e. ten values from each energy vector chosen randomly along the path $\Gamma - K$). The latter has certified and shown that the artificial neural network is a pragmatic model that would allow performing and improving interesting tasks in terms of accuracy and computation time as seen in Figure 6 and Tables 2 and 3.

Successful training and validation proved the credibility of the neural network approach where the output quantities calculated by the network were in excellent agreement with the values given in refs [19,20]. The

network topology, chosen to expect the quantities of interest of the transition metals, included the data of energy bands as an input unit; moreover, the data set taken from Table 1 was used as an output unit of the network. A four-layer neural network was chosen as the architecture of the model. The initialization of the network weights and bias were generated by using Nguyen and Widrow's initialization algorithm, and the neural network was trained with the Levenberg–Marquardt back propagation algorithm. Finally, the neurocomputing model can be considered universal in nature since it can be extended as a new approximation method used in complex systems especially to the extent it that is useful without any constraints on the number of components that can be studied.

References

[1] Srinivasan, S.; Saghir, M. Z. *Neural Computing and Applications* **2012**, *24*, 287-299.

[2] Srinivasan, S.; Saghir, M. Z. *Neural Computing and Applications* **2014**, *25*, 1193-1203.

[3] Bourouis, C.; Meddour, A.; Moussaoui, A. K. *J. Mol. Struct: Theochem.* **2006**, *777*, 45-51.

[4] Hornik, K.; Stinchcombe, M. *Neural Networks* **1989**, *2*, 359-366.

[5] Dreyfus, G.; Martinez, J. M.; Samulides, M.; Gordon, M. B.; Badran, F.; Thiria, S. *Neural Networks, Methodology and Applications*. Eyrolles; Paris, France, 2004.

[6] LeCun, Y.; Bottou, L.; Bengio, I.; Haffner, P. *Intelligent Signal Processing. IEE Press.* **2001**, 306-351.

[7] Ventresca, M.; Tizhoosh, H. R. *In Proceedings of the International Joint Conference on Neural Networks. IEEE Press,* Piscataway, NJ, USA. **2009**.

[8] Oussar, Y.; Dreyfus, G. *Neurocomputing.* **2000**, *34*, 131-143.

[9] Hagan, M. T.; Menhaj, M. B. *IEEE Trans. Neural Netw.* **1994**, *5*, 989-993.

[10] Cybenko, G. Math. *Control Signals Syst.* **1988**, *2*, 303-314.

[11] Srinivasan, S.; Saghir, M. Z. *Neural Computing and Applications* **2013**, *37*, 2850-2869

[12] Liu, B.; Xin, Y. U.; Cheung, R. C.; Yan, A. H. *Neurocomputing* **2014**, *134*, 239-246.

[13] Ramasamy, P.; Chandel S. S; Yadav, A. K. *Renewable Energy* **2015**, *80*, 338-347.

[14] Gusynin, V. P.; Sharapov, S. G.; Carbotte, J. P. *Int. J. Mod. Phys. B.* **2007**, *21*, 4611-4657.

[15] Smith, D.; Smekal, L. V. *Phys. Rev. B.* **2014**, *89*, 195-123.

[16] McEniry, E. J.; Madsen, G. K. H.; Drain, J. F.; Drautz, R. *J. Phys. Condens Matter.* **2011**, *23*, 276004.

[17] Kaxiras, E. *Atomic and Electronic Structure of Solids,* Cambridge University Press: New York, NY, USA, 2013.

[18] Slater, J. C., Koster, G. F. *Phys. Rev.* **1954**, *94*, 1498-1524.

[19] Harrison, W. A. *Elementary Electronic Structure.* World Scientific: Singapore, 2004.

[20] Kittel, C. *Introduction to Solid State Physics*, 8th ed.; Wiley; New York, NY, USA, 2008.

[21] McCulloch, W. S.; Pitts, W. *Bulletin Math. Biophys.* **1943**, *5*, 115-133.

[22] Hum, F. M.; Kostanic, I. *Principles of Neurocomputing for Science and Engineering.* 2nd ed.; McGraw Hill: New York, NY, USA, 2001.

[23] Levenberg, K. *Quarterly J. Appl. Math.* **1944**, *2*, 164-168.

[24] Marquardt, D. W. *SIAM Journal on Optimization.* **1963**, *11*, 431-441.

[25] Nguyen, D.; Widrow, B. *Proceedings of the International Joint Conference on Neural Networks.* June 1990.