Supplementary Material

Table S1. Botanical identification of open-grown-trees in the Cerrado Biome.

Botanical Family/Scientific Name	Tree Frequency	Crown Geometric Shapes
Anacardiaceae		
Astronium graveolens Jacq.	1	3
Lithraea molleoides (Vell.) Engl.	1	1
Tapirira guianensis Aubl.	3	2
Annonaceae		
Annona montana Macfad.	4	2, 4
Apocynaceae		
Aspidosperma macrocarpon Mart.	1	2
Aspidosperma parvifolium A.DC.	4	2
Aspidosperma sp.	1	2
Hancornia speciosa Gomes	2	1
Araliaceae		
Didymopanax macrocarpus (Cham. and Schltdl.) Frodin	1	3
Asteraceae		
Piptocarpha rotundifolia (Less.) Baker	1	3
Bignoniaceae		
Cybistax antisyphilitica (Mart.) Mart.	1	5
Handroanthus albus (Cham.) Mattos	6	2, 4, 5
Handroanthus avellanedae (Lorentz ex Griseb.) Mattos	2	2, 3
Zeyheria tuberculosa (Vell.) Bureau ex Verl.	4	2, 4
Boraginaceae		
cordia trichotoma (Vell.) Arráb. ex Steud.	3	5
Caryocaraceae		
Caryocar brasiliense Cambess.	13	2, 3, 4, 5
Chrysobalanaceae		
Licania apetala (E.Mey.) Fritsch	1	2
Combretaceae		
Terminalia argentea Mart. and Zucc.	3	2
Terminalia glabrescens Mart.	2	3
Ebenaceae		
Diospyros lasiocalyx (Mart.) B.Walln.	1	2
Fabaceae		
Anadenanthera macrocarpa (Benth.) Brenan	6	2, 3
Bowdichia virgilioides Kunth	17	2, 3, 4
Copaifera langsdorffii Desf.	3	1, 2, 3

Dimorphandra mollis Benth.	2	2
Enterolobium gummiferum (Mart.) J.F.Macbr.	2	3
Hymenaea stigonocarpa Mart. ex Hayne	3	2
Leptolobium dasycarpum Vogel	3	1, 3
Leptolobium elegans Vogel	2	3
Plathymenia reticulata Benth.	11	2, 3, 4
Senna multijuga (Rich.) H.S.Irwin and Barneby	1	1
Stryphnodendron adstringens (Mart.) Coville	1	2
Lamiaceae		
Aegiphila integrifolia Cham.	2	2
Lauraceae		
Ocotea spixiana (Nees) Mez	2	2, 3
Lecythidaceae		
Cariniana estrellensis (Raddi) Kuntze	1	2
Loganiaceae		
Strychnos pseudoquina A.St.-Hil.	1	2
Lythraceae		
Lafoensia pacari A.St.-Hil.	5	2, 3, 4
Malpighiaceae		
Byrsonima coccolobifolia Kunth	1	2
Malvaceae		
Ceiba speciosa (A.St.-Hil.) Ravenna	1	2
Eriotheca gracilipes (K.Schum.) A.Robyns	5	2, 3, 4
Eriotheca pubescens (Mart. and Zucc.) Schott and Endl.	1	2
Luehea divaricata Mart. and Zucc.	1	4
Pseudobombax tomentosum (Mart.) A.Robyns	1	1
Meliaceae		
Cedrela fissilis Vell.	3	2, 4
Moraceae		
Maclura tinctoria (L.) D.Don ex Steud.	4	1, 2
Myrtaceae		
Eugenia dysenterica (Mart.) DC.	1	1
Myrcia tomentosa (Aubl.) DC.	1	1
Primulaceae		
Myrsine gardneriana A.DC.	11	1, 2, 4, 5
Rhamnaceae		
Rhamnidium elaeocarpum Reissek	5	1, 2
Rutaceae		
Zanthoxylum rhoifolium Lam.	3	2, 4
Zanthoxylum riedelianum Engl.	1	2

Sapindaceae		
Matayba guianensis Aubl.	2	2, 4

Sapotaceae		
Chrysophyllum marginatum (Hook. and Arn.) Radlk.	4	1, 2, 3, 5
Pouteria ramiflora (Mart.) Radlk.	1	2
Solanaceae		
Solanum lycocarpum A.St.-Hil.	1	1
Urticaceae		
Cecropia pachystachya Trécul	1	3
Vochysiaceae		
Qualea grandiflora Mart.	25	1, 2, 3, 4, 5
Qualea multiflora Mart.	1	3
Qualea parviflora Mart.	4	2, 3, 4
-		
IN	5	1, 2, 4
Total	200	

Table S2. Growth space considering the size of the potential crown diameter in open-growing trees.

D_{eq}	CD	GS	N	G
5	6.2	0.0030	388	0.76
10	7.3	0.0042	275	2.16
15	8.4	0.0056	207	3.66
20	9.5	0.0071	162	5.10
25	10.6	0.0088	131	6.44
30	11.6	0.0106	109	7.69
35	12.6	0.0125	92	8.87
40	13.6	0.0145	79	9.98
45	14.6	0.0166	69	11.03
50	15.5	0.0188	61	12.04
55	16.4	0.0211	55	13.03
60	17.2	0.0233	49	13.98
65	18.1	0.0257	45	14.93
70	18.9	0.0280	41	15.86
75	19.7	0.0304	38	16.79
80	20.4	0.0327	35	17.73
85	21.1	0.0351	33	18.67
90	21.8	0.0374	31	19.62
95	22.5	0.0398	29	20.58
100	23.1	0.0421	27	21.57
105	23.7	0.0443	26	22.57
110	24.3	0.0465	25	23.60
115	24.9	0.0486	24	24.66
120	25.4	0.0507	23	25.76
125	25.9	0.0527	22	26.89
130	26.4	0.0546	21	28.07
135	26.8	0.0564	20	29.28
140	27.2	0.0582	20	30.55
145	27.6	0.0598	19	31.87
150	28.0	0.0614	19	33.25
155	28.3	0.0628	18	34.69
160	28.6	0.0641	18	36.20
165	28.8	0.0653	18	37.79
170	29.1	0.0664	17	39.46
175	29.3	0.0674	17	41.21
180	29.5	0.0682	17	43.06
185	29.6	0.0690	17	45.02
190	29.8	0.0695	17	47.08

195	29.9	0.0700	16	49.27
200	29.9	0.0703	16	51.59
205	30.0	0.0705	16	54.06
210	30.0	0.0706	16	56.68
215	30.0	0.0705	16	59.48
220	29.9	0.0703	16	62.46
225	29.8	0.0699	17	65.66

Modeling using artificial neural networks (A_{NNs})

Multi-layer perceptron (MLP) A_{NNs} (Figure S1) with only one hidden layer were used to train the data [107]. The starting from the data normalization according to two types of intervals [0;1] according to Equation (S1).

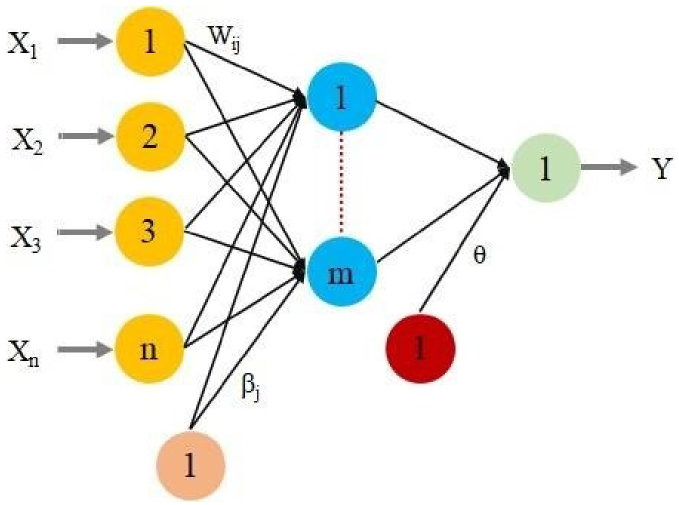

Figure S1. Basic structure of an MLP.

Therefore, it was considered that Y = estimation of the value of the dependent variable; X_i = input value of the i-th independent variable; w_{ij} = connection weight between the i-th input neuron and the j-th neuron of the hidden layer; β_j = bias value of the j-th neuron of the hidden layer; v_j = connection weight between the j-th neuron of the hidden layer and the output neuron; θ = bias value of the output neuron. After, Equation S2 was applied.

$$\Delta = \frac{UL - IL}{X_{maximum} - X_{minimum}} \quad (S1)$$

$$X_{equal} = IL - \Delta \times X_{minimum} + \Delta \times X_i \quad (S2)$$

where X_i is the value to be equalized, $X_{minimum}$ is the lowest value of the dataset, $X_{maximum}$ is the highest value of the dataset, UL is upper limit, and IL is inferior limit.

The modeling was considered the development of six models of artificial neural networks (CD [A_{NN1}],.., CD [A_{NN6}]), with the input variables inserted in the sequence [X_1 = Deq (cm); X_2 = D_{03} (cm);

$X_3 = C_L$ (m); $X_4 = D_{07}$ (cm); $X_5 = C_{BH}$ (m); $X_6 = C_{GS}$]. For each input variable, the insertion of three neurons in the hidden layer (m) was considered. Thus, the CD [A_{NN6}] model presented 18 neurons in the hidden layer (m = 18) (see Table S3). In the output layer (Y), the value of the crown diameter (CD) was obtained, in meters. The training the networks, the logistic activation function (Equation S3) was used in the hidden layer and linear (Equation S4) in the output layer.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (S3)$$

$$g(x) = x \quad (S4)$$

The sigmoid activation function has this name because its curve resembles the letter "S"; this function returns only positive values, in a range from 0 to 1 [108]. The identity activation function has greater simplicity compared to the others and has a linear output.

The A_{NNs} prediction is possible through the mathematical expression described for MLP (Equation S5), as follows:

$$Y = g\left(\theta + \sum_{j=1}^{m} v_j \left[\sum_{i=1}^{n} f(w_{ij}X_i + \beta_j)\right]\right) \quad (S5)$$

where Y = estimation of the value of the dependent variable; X_i = input value of the i-th independent variable; w_{ij} = connection weight between the i-th input neuron and the j-th neuron of the hidden layer; β_j = bias value of the j-th neuron of the hidden layer; v_j = connection weight between the j-th neuron of the hidden layer and the output neuron; θ = bias value of the output neuron; *f*(.) = hidden layer activation function; *g*(.) = output activation function.

Table S3. Parameters (synaptic weights and biases) of the A_{NNs} selected to describe the crown diameter of open-grown-trees in the Cerrado Biome

Description	Simbology	CD [A_{NN1}]	CD [A_{NN2}]	CD [A_{NN3}]	CD [A_{NN4}]	CD [A_{NN5}]	CD [A_{NN6}]
		Parameters*					
	w_{11}	-52.285701	0.441175	-0.368468	0.788746	0.352935	0.008867
	w_{12}	6.067180	0.024614	-1.308982	-29.926575	1.197814	-1.283944
	w_{13}	-2.556499	-1.122738	2.948249	-1.614895	-0.018677	-3.375115
	w_{14}		5.499550	-3.962290	1.266468	1.480785	1.776070
	w_{15}		1.054599	-6.307936	-0.954370	-1.130997	-0.844790
	w_{16}		1.545627	-1.003498	0.934919	-0.285297	-0.234891
	w_{17}			-3.536656	-1.251683	0.801147	-1.006638
	w_{18}			-27.327223	0.645551	1.549380	0.757662
	w_{19}			-0.462741	-0.382814	0.108774	0.729109
	w_{110}				-1.091274	2.783036	-0.905680
	w_{111}				1.259159	-28.835061	-1.631331
	w_{112}				3.237491	0.968321	-0.092664
	w_{113}					0.236701	-1.008296
	w_{114}					-0.569685	-0.525780
	w_{115}					-0.232053	0.376922
	w_{116}						14.383766
	w_{117}						-2.366025
	w_{118}						1.119998
Connection weight between the i-th input neuron and the j-th neuron of the hidden layer	w_{21}		1.240771	0.152024	-1.565146	-0.016037	-0.390417
	w_{22}		1.986934	-0.270411	12.241726	-0.443947	0.537074
	w_{23}		-5.836951	-0.956231	-0.620027	-1.558500	-4.866437
	w_{24}		2.251971	6.144645	1.256208	1.198452	0.191175
	w_{25}		7.687187	-43.095923	0.153504	-1.199291	0.506829
	w_{26}		0.230043	-0.628444	-2.613559	-1.365705	-0.052260
	w_{27}			3.997805	18.295995	0.694334	0.707511
	w_{28}			8.067229	1.977717	2.191541	1.902660
	w_{29}			-0.212183	-1.038435	0.148147	-0.113810
	w_{210}				0.034761	-0.887761	-0.794508
	w_{211}				1.873751	5.540455	0.866439
	w_{212}				0.039941	-0.366681	-0.992417
	w_{213}					-0.646160	-1.550012
	w_{214}					-0.424988	-0.194124
	w_{215}					-0.593907	1.949137
	w_{216}						-3.168511
	w_{217}						0.568626
	w_{218}						0.753377
	w_{31}			-1.058894	-1.893249	-0.089171	-0.173472
	w_{32}			-0.441188	17.934741	0.034670	1.294303

w_{33}	-0.301711	-0.203926	2.352748	4.318236
w_{34}	-4.762883	0.302595	0.853023	0.017615
w_{35}	-0.542065	-1.697443	2.196633	0.268517
w_{36}	0.114667	0.780587	2.813368	-1.678827
w_{37}	3.546779	-9.598029	-1.160846	-0.089845
w_{38}	6.870947	0.488596	1.815828	0.448170
w_{39}	-0.732768	-0.891396	0.917626	-0.396973
w_{310}		-1.919412	0.701266	-1.225497
w_{311}		-2.229371	12.791242	0.815772
w_{312}		0.451052	-25.805858	-0.358592
w_{313}			-1.485516	-9.838083
w_{314}			-0.764018	0.519500
w_{315}			0.759926	0.597432
w_{316}				-6.922471
w_{317}				0.605203
w_{318}				-0.595918
w_{41}		-0.436547	0.342353	-1.428986
w_{42}		7.658021	-0.377858	0.054484
w_{43}		-0.526418	0.027108	-7.129777
w_{44}		-0.296933	0.331817	-1.991782
w_{45}		-1.415039	0.751963	1.602134
w_{46}		-1.786652	0.244533	-0.597560
w_{47}		48.255409	0.135645	0.101074
w_{48}		-0.636502	0.854073	-0.939687
w_{49}		-4.150908	0.871344	0.757750
w_{410}		1.995379	0.437633	-0.892106
w_{411}		-0.378686	-30.069369	0.929289
w_{412}		-0.225500	31.231198	0.140579
w_{413}			0.773047	10.245140
w_{414}			0.081473	-2.173056
w_{415}			0.482071	0.442091
w_{416}				-8.103308
w_{417}				-0.894157
w_{418}				-0.595494
w_{51}			0.897456	1.249620
w_{52}			-0.928073	-1.066804
w_{53}			0.469211	-1.638531
w_{54}			0.749753	-0.890046
w_{55}			-0.674494	-0.593003
w_{56}			-1.081294	-0.846941
w_{57}			-1.774760	0.670528
w_{58}			-2.417113	-1.261176

	w_{59}					0.180632	0.290387
	w_{510}					0.858854	-1.068239
	w_{511}					-7.161013	0.293048
	w_{512}					10.555245	-1.234492
	w_{513}					-0.139305	8.808620
	w_{514}					-0.252607	-0.297507
	w_{515}					-0.441234	0.662599
	w_{516}						-0.991709
	w_{517}						-1.993560
	w_{518}						1.610218
	w_{61}						-1.253547
	w_{62}						-0.351199
	w_{63}						1.462390
	w_{64}						-0.485148
	w_{65}						0.772739
	w_{66}						0.793475
	w_{67}						-0.049888
	w_{68}						1.444210
	w_{69}						-1.316608
	w_{610}						1.646875
	w_{611}						-0.295740
	w_{612}						0.976105
	w_{613}						2.545005
	w_{614}						-0.378622
	w_{615}						-1.582336
	w_{616}						-0.551052
	w_{617}						0.398931
	w_{618}						-0.862134
Bias value of the j-th neuron of the hidden layer	β_1	10.106684	-1.409141	-0.573714	-0.228742	-0.749953	0.853635
	β_2	-0.806814	-1.521088	0.416304	-1.586991	0.557637	0.556086
	β_3	1.998134	-0.638708	0.603434	0.691158	0.079495	-2.755697
	β_4		-1.304119	2.367323	-0.608529	-0.565994	1.326871
	β_5		-2.130729	33.999751	1.112479	-0.684473	-0.020924
	β_6		-1.689605	-1.011302	-0.787537	1.132387	-0.046967
	β_7			1.060184	-3.075630	0.604845	-1.654833
	β_8			-2.680666	-0.832502	-1.195707	-0.481469
	β_9			-0.469019	0.016202	-1.347570	0.996778
	β_{10}				-0.771936	0.095406	0.798270
	β_{11}				-0.373772	2.937064	0.036856
	β_{12}				0.708892	-0.358073	0.433929
	β_{13}					1.329899	-0.128215
	β_{14}					0.515573	-1.353280

	β_{15}					0.874171	0.574683
	β_{16}						-1.594957
	β_{17}						-0.229665
	β_{18}						-0.986161
Connection weights	v_1	0.100666	0.601329	-0.246043	1.453558	-0.774367	0.303532
	v_2	0.980560	1.064435	-0.356684	0.144953	-1.207625	1.292379
	v_3	-0.769734	-0.167223	0.220761	-0.562051	1.889052	-0.890050
	v_4		1.029324	-0.170448	0.101620	-0.881039	0.668162
	v_5		-0.717272	-0.249255	-0.406641	-1.831858	0.190370
	v_6		0.137631	-0.734862	0.485442	-1.136634	0.152851
	v_7			0.766223	0.100875	1.337729	-1.351368
	v_8			-0.817319	1.183491	0.260184	0.695761
	v_9			-0.042120	0.238218	1.128355	0.044892
	v_{10}				-0.513955	0.710968	0.014531
	v_{11}				0.151038	-0.139021	-0.291733
	v_{12}				0.741138	-0.141518	-0.676431
	v_{13}					-0.360573	-0.288902
	v_{14}					-1.071674	-0.945386
	v_{15}					1.255069	-1.188071
	v_{16}						-0.683157
	v_{17}						-1.112205
	v_{18}						1.337282
Bias value of the output neuron	θ	0.295600	-0.395647	0.261993	-1.047888	0.559285	0.151516

*Input variables used were as follows: D_{eq} is diameter measured 1.3 m; D_{03} is diameter measured 0.3 m; C_L is crown length; D_{07} is diameter measured 0.7 m; C_{BH} is crown base height; and C_{GS} is crown geometric shapes.

Code in R

```
weights = structure(list(
ANN1 = c(-52.285701, 6.06718, -2.556499, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
            10.106684, -0.806814, 1.998134, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0.100666, 0.98056, -0.769734, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            0.2956),
ANN2 = c(0.441175, 0.024614, -1.122738, 5.49955, 1.054599, 1.545627, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.240771,
 1.986934, -5.836951, 2.251971, 7.687187, 0.230043, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            -1.409141, -1.521088, -0.638708, -1.304119, -2.130729, -1.689605, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0.601329, 1.064435, -0.167223, 1.029324, -0.717272, 0.137631, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            -0.395647),
ANN3 = c(-0.368468, -1.308982, 2.948249, -3.96229, -6.307936, -1.003498, -3.536656, -27.327223, -0.462741, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.152024, -
0.270411, -0.956231, 6.144645, -43.095923, -0.628444, 3.997805, 8.067229, -0.212183, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1.058894, -0.441188, -0.301711, -
4.762883, -0.542065, 0.114667, 3.546779, 6.870947, -0.732768, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             -0.573714, 0.416304, 0.603434, 2.367323, 33.999751, -1.011302, 1.060184, -2.680666, -0.469019, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             -0.246043, -0.356684, 0.220761, -0.170448, -0.249255, -0.734862, 0.766223, -0.817319, -0.04212, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.261993),
ANN4 = c(0.788746, -29.926575, -1.614895, 1.266468, -0.95437, 0.934919, -1.251683, 0.645551, -0.382814, -1.091274, 1.259159, 3.237491, 0, 0, 0,
0, 0, 0, -1.565146, 12.241726, -0.620027, 1.256208, 0.153504, -2.613559, 18.295995, 1.977717, -1.038435, 0.034761, 1.873751, 0.039941, 0, 0, 0, 0,
0, 0, -1.893249, 17.934741, -0.203926, 0.302595, -1.697443, 0.780587, -9.598029, 0.488596, -0.891396, -1.919412, -2.229371, 0.451052, 0, 0, 0, 0, 0,
0, -0.436547, 7.658021, -0.526418, -0.296933, -1.415039, -1.786652, 48.255409, -0.636502, -4.150908, 1.995379, -0.378686, -0.2255, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            -0.228742, -1.586991, 0.691158, -0.608529, 1.112479, -0.787537, -3.07563, -0.832502, 0.016202, -0.771936, -0.373772, 0.708892, 0, 0, 0,
0, 0, 0,
            1.453558, 0.144953, -0.562051, 0.10162, -0.406641, 0.485442, 0.100875, 1.183491, 0.238218, -0.513955, 0.151038, 0.741138, 0, 0, 0, 0, 0,
0,
            -1.047888),
ANN5 = c(0.352935, 1.197814, -0.018677, 1.480785, -1.130997, -0.285297, 0.801147, 1.54938, 0.108774, 2.783036, -28.835061, 0.968321, 0.236701,
-0.569685, -0.232053, 0, 0, 0, -0.016037, -0.443947, -1.5585, 1.198452, -1.199291, -1.365705, 0.694334, 2.191541, 0.148147, -0.887761, 5.540455, -
0.366681, -0.64616, -0.424988, -0.593907, 0, 0, 0, -0.089171, 0.03467, 2.352748, 0.853023, 2.196633, 2.813368, -1.160846, 1.815828, 0.917626,
0.701266, 12.791242, -25.805858, -1.485516, -0.764018, 0.759926, 0, 0, 0, 0.342353, -0.377858, 0.027108, 0.331817, 0.751963, 0.244533, 0.135645,
0.854073, 0.871344, 0.437633, -30.069369, 31.231198, 0.773047, 0.081473, 0.482071, 0, 0, 0, 0.897456, -0.928073, 0.469211, 0.749753, -0.674494, -
   1.081294, -1.77476, -2.417113, 0.180632, 0.858854, -7.161013, 10.555245, -0.139305,
               -0.252607, -0.441234, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
            -0.749953, 0.557637, 0.079495, -0.565994, -0.684473, 1.132387, 0.604845, -1.195707, -1.34757, 0.095406, 2.937064,
-0.358073, 1.329899, 0.515573, 0.874171, 0, 0, 0,
             -0.774367, -1.207625, 1.889052, -0.881039, -1.831858, -1.136634, 1.337729, 0.260184, 1.128355, 0.710968, -0.139021, -0.141518, -
0.360573, -1.071674, 1.255069, 0, 0, 0,
            0.559285),
ANN6 = c(0.008867, -1.283944, -3.375115, 1.77607, -0.84479, -0.234891, -1.006638, 0.757662, 0.729109, -0.90568, -1.631331, -0.092664, -
1.008296, -0.52578, 0.376922, 14.383766, -2.366025, 1.119998, -0.390417, 0.537074, -4.866437, 0.191175, 0.506829, -0.05226, 0.707511, 1.90266, -
0.11381,
-0.794508, 0.866439, -0.992417, -1.550012, -0.194124, 1.949137, -3.168511, 0.568626, 0.753377, -0.173472, 1.294303, 4.318236,
0.017615, 0.268517, -1.678827, -0.089845, 0.44817, -0.396973, -1.225497, 0.815772, -0.358592, -9.838083, 0.5195, 0.597432,
-6.922471, 0.605203, -0.595918, -1.428986, 0.054484, -7.129777, -1.991782, 1.602134, -0.59756, 0.101074, -0.939687, 0.75775, -0.892106, 0.929289,
0.140579, 10.24514, -2.173056, 0.442091, -8.103308, -0.894157, -0.595494, 1.24962, -1.066804, -1.638531,
            -0.890046, -0.593003, -0.846941, 0.670528, -1.261176, 0.290387, -1.068239, 0.293048, -1.234492, 8.80862, -0.297507, 0.662599,
-0.991709, -1.99356, 1.610218, -1.253547, -0.351199, 1.46239, -0.485148, 0.772739, 0.793475, -0.049888, 1.44421, -1.316608,
1.646875, -0.29574, 0.976105, 2.545005, -0.378622, -1.582336, -0.551052, 0.398931, -0.862134,
            0.853635, 0.556086, -2.755697, 1.326871, -0.020924, -0.046967, -1.654833, -0.481469, 0.996778, 0.79827, 0.036856, 0.433929, -
0.128215, -1.35328, 0.574683, -1.594957, -0.229665, -0.986161,
            0.303532, 1.292379, -0.89005, 0.668162, 0.19037, 0.152851, -1.351368, 0.695761, 0.044892, 0.014531, -0.291733, -0.676431, -0.288902, -
0.945386, -1.188071, -0.683157, -1.112205, 1.337282,
            0.151516)),
row.names = c("w11", "w12", "w13", "w14", "w15", "w16", "w17", "w18", "w19", "w110", "w111", "w112", "w113", "w114", "w115", "w116",
"w117", "w118", "w21", "w22", "w23", "w24", "w25", "w26", "w27", "w28", "w29", "w210", "w211", "w212", "w213", "w214", "w215", "w216",
"w217", "w218", "w31", "w32", "w33", "w34", "w35", "w36", "w37", "w38", "w39", "w310", "w311", "w312", "w313", "w314", "w315", "w316",
"w317", "w318", "w41", "w42", "w43", "w44", "w45", "w46", "w47", "w48", "w49", "w410", "w411", "w412", "w413", "w414", "w415", "w416",
"w417", "w418", "w51", "w52", "w53", "w54", "w55", "w56", "w57", "w58", "w59", "w510", "w511", "w512", "w513", "w514", "w515", "w516",
"w517", "w518", "w61", "w62", "w63", "w64", "w65", "w66", "w67", "w68", "w69", "w610", "w611", "w612", "w613", "w614", "w615", "w616",
"w617", "w618",
            "b1", "b2", "b3", "b4", "b5", "b6", "b7", "b8", "b9", "b10", "b11", "b12", "b13", "b14", "b15", "b16", "b17", "b18",
            "v1", "v2", "v3", "v4", "v5", "v6", "v7", "v8", "v9", "v10", "v11", "v12", "v13", "v14", "v15", "v16", "v17", "v18",
            "teta"), class = "data.frame")
ann = function(input){
    n = length(input)
    length(input) = 6
    input[is.na(input)] = 0
    norma_values = structure(list(minimum = c(5.1,5.9,0.8,5.6,1,1,1.36), maximum = c(226,226,23.8,226,9.6,5,29.8)),
    class = "data.frame", row.names = c('input1','input2','input3','input4','input5','input6','output'))
    norma_input = c()
```

```
for(i in seq(1,6)){
     DeltaInput = (1-0)/(norma_values[paste('input',i,sep=''),'maximum']-norma_values[paste('input',i,sep=''),'minimum'])
norma = 0 - DeltaInput*norma_values[paste('input',i,sep=''),'minimum']+DeltaInput*input[i]
norma_input = append(norma_input,norma)
              }
#weights
ann = paste('ANN',n,sep='')
total_summation = c()
for(i in seq(1,n*3)){
summation = (weights[paste('v',i,sep=''),ann]*
   (((1/(1+exp(-(weights[paste('w1',i,sep=''),ann]*norma_input[1]+weights[paste('w2',i,sep=''),ann]*norma_input[2]+
             weights[paste('w3',i,sep=''),ann]*norma_input[3]+weights[paste('w4',i,sep=''),ann]*norma_input[4]+
             weights[paste('w5',i,sep=''),ann]*norma_input[5]+weights[paste('w6',i,sep=''),ann]*norma_input[6]+
             weights[paste('b',i,sep=''),ann])))))))

total_summation = append(total_summation,summation)
              }
 g=sum(total_summation)+weights['teta',ann]
 DeltaOutput=(1-0)/(norma_values['output','maximum']-norma_values['output','minimum'])
 Output=(g-0+DeltaOutput*norma_values['output','minimum'])/DeltaOutput
 return(Output)
              }
#input --> [X1 = Deq; X2 = D03; X3 = CL; X4 = D07; X5 = CBH; X6 = CGS]
# ANN1
round(ann(c(6.0)),2)
# ANN2 … ANN5
# ANN6
round(ann(c(6.0,7.2,2.8,6.4,1.2,4)),2)
```

References

107. Haykin, S. *Redes Neurais - Princípios e Prática*; Engel, P.M., Ed.; 2nd ed.; Bookman: Porto Alegre, 2001; Vol. 7659; ISBN 85-7307-718-2.

108. Heaton, J. *Introduction to the Math of Neural Networks*; WordsRU.com, Ed.; 1st ed.; Heaton Research, Inc, 2012; ISBN 978-1475190878.