inventions

MDPI

Article

Sensitivity Analysis of Artificial Neural Networks Identifying JWH Synthetic Cannabinoids Built with Alternative Training Strategies and Methods

Catalina Mercedes Burlacu, Adrian Constantin Burlacu and Mirela Praisler *

Department of Chemistry, Physics and Environment, Faculty of Science and Environment, "Dunarea de Jos" University of Galati, 47 Domneasca Street, 800008 Galati, Romania
* Correspondence: mirela.praisler@ugal.ro

Abstract: This paper presents the alternative training strategies we tested for an Artificial Neural Network (ANN) designed to detect JWH synthetic cannabinoids. In order to increase the model performance in terms of output sensitivity, we used the *Neural Designer* data science and machine learning platform combined with the programming language *Python*. We performed a comparative analysis of several optimization algorithms, error parameters and regularization methods. Finally, we performed a new goodness-of-fit analysis between the testing samples in the data set and the corresponding ANN outputs in order to investigate their sensitivity. The effectiveness of the new methods combined with the optimization algorithms is discussed.

Keywords: JWH synthetic cannabinoids; artificial neural networks; optimization algorithms

Citation: Burlacu, C.M.; Burlacu, A.C.; Praisler, M. Sensitivity Analysis of Artificial Neural Networks Identifying JWH Synthetic Cannabinoids Built with Alternative Training Strategies and Methods. *Inventions* **2022**, *7*, 82. https://doi.org/10.3390/inventions7030082

Academic Editor: Anastasios Doulamis

Received: 13 July 2022
Accepted: 29 August 2022
Published: 13 September 2022

1. Introduction

Artificial neural networks (ANN) contain a set of parameters that can be adjusted to perform different tasks. These structures have universal approximation properties, which means that they can approximate any function in any size and, generally, up to a desired degree of accuracy [1–4].

In this article, we present a series of deep learning training and optimization strategies that have been applied to improve the performance of an ANN identifying JWH-synthetic-cannabinoid-class membership. In order to increase the system sensitivity, we trained and optimized an initial model on four new architectures. For this purpose, we used the data science and machine learning platform *Neural Designer.* The best version was implemented in the *Python 3.10* programming language for further development and improvement.

The classification efficiencies (output results) obtained for several combinations of algorithms, error parameters and regularization methods were compared. The good fit between the test samples and the corresponding ANN outputs was also analyzed. The effectiveness of the methods was analyzed and is presented in detail.

2. Materials and Methods

The initial input database of 150 synthetic chemicals included JWH synthetic cannabinoids, others synthetic cannabinoids and other substances of abuse. These designer drugs were divided into three classes referred to as "Class 1—JWH", "Class 2—non-JWH Cannabinoids" and "Class 3—Others". The group of positives contained 50 JWH synthetic cannabinoids, while the group of negatives included 100 compounds, i.e., 50 non-JWH cannabinoids and 50 other substances of forensic interest [5].

We used the quantitative structure–activity relationship (QSAR) method to estimate and predict the pharmacokinetics, drug-likeness and medicinal chemistry friendliness of each input compound by calculating a number of 300 molecular descriptors in terms of their physical and chemical properties, as well as 50 indices characterizing their chemical

absorption, distribution, metabolism, excretion and toxicity activity (ADMET). The descriptors were selected from three blocks, i.e., topological, 3D-MoRSE (molecule representation of structure based on electron diffraction) and toxicity [6].

Only the first 150 most relevant descriptors were selected and used for the final computational and modelling stage. Hence, the input database was a matrix consisting of 150 samples $\times$ 150 variables. The shape, feature and target types of the data set, including the list of the computed and tested input molecular descriptors was presented in a previously published article [7].

The data set was divided into three subsets of samples: training, selection and testing. Hence, we used 90 training samples (60%), 30 selection samples (20%) and 30 testing samples (20%). To discover redundancies between the input variables, we used a correlation matrix, which represents a numerical value between -1 and 1 that expresses the strength of the relationship between two variables [8]. The types of layers the most frequently used in our classification model were the perceptron layer, the probabilistic layer and the scaling and bounding layers. The objective of the selection was to find the best-performing network architecture in terms of system sensitivity.

To avoid underfitting and overfitting, the neuron selection algorithm responsible for finding the optimal number of neurons in the networks was the Growing neurons algorithm [9]. The *Neural Designer* data science and machine learning platform was used to generate the mathematical expression represented by ANNs in order to export and incorporate them into the programming language *Python 3.10* in the so-called production mode.

Our general training strategy consisted of two different concepts, i.e., the loss index and the optimization algorithms. The error was the essential term in the loss expression. The most important errors that we estimated were the sum squared error, the mean squared error, the root mean squared error, the normalized squared error and the Minkowski error. We used the L1 and L2 regularization methods, which involve the sum of the absolute values of all parameters and the square sum of all the parameters in the ANN. The loss index was measured on the data set and could be represented as a hyper-surface with the parameters as coordinates (see Figure 1) [10].

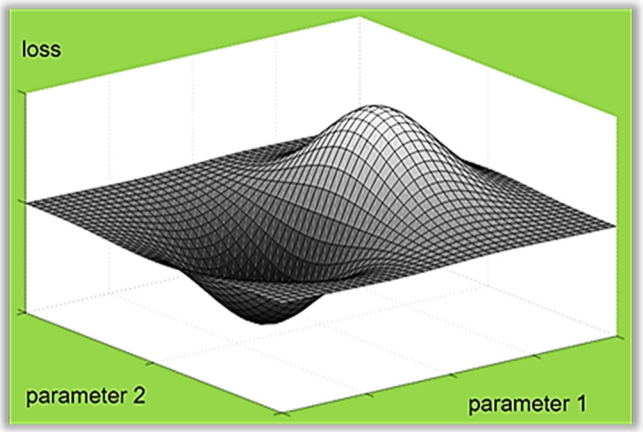

Figure 1. Loss index viewed as a hyper-surface with parameters as coordinates.

In order to train the ANN, we generated a sequence of parameter vectors so that the loss index was reduced at each iteration of the algorithm.

3. Results

Five different optimization algorithms were used and compared, each with a variety of different calculation and storage requirements: gradient descent [11], conjugate gradient, quasi-Newton method, Levenberg-Marquardt algorithm [12] and adaptative linear momentum [13].

In order to scale the inputs, we calculated the following parameters: the minimum, the maximum, the mean and the standard deviation (see Table 1). The ANN architecture is presented in Figure 2 for version 1. The architectures of the following versions (2, 3 and 4) were also with one hidden layer perceptrons and had the same input and output layers as version 1. On the other hand, their hidden layers contained three (version 2), one (version 3) and six (version 4) nodes respectively.

Table 1. Values used for scaling the inputs for all ANN versions.

Input	Minimum	Maximum	Mean	Deviation	Scaler
Topological	4.300000	7.900000	5.843330	0.828066	Mean St. Dev.
3D-MoRSE	2.000000	4.400000	3.054000	0.433594	Mean St. Dev.
ADMET	1.000000	6.900000	3.758670	1.764420	Mean St. Dev.

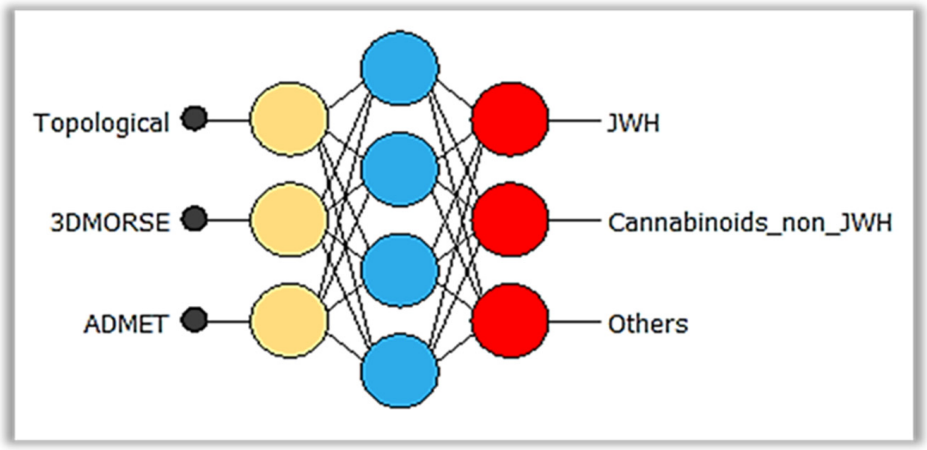

Figure 2. Network architecture, version 1: scaling layer (yellow), perceptron layer (blue), probabilistic layer (red).

We used the adaptive moment estimation (version 1), the Levenberg–Marquardt (version 2), the gradient descent (version 3) and the conjugate gradient optimization algorithms, as well as the growing neuron selection (all versions) method with the L1 (versions 2, 3 and 4) and L2 (version 1) regularization methods.

4. Discussion

The confusion matrices, calculated for each architecture and 30 testing samples, are presented in Tables 2–5 and the error results are highlighted in Table 6. The activation functions used were the hyperbolic tangent (version 1), the rectified linear (versions 2, 3 and 4) and Softmax (all versions).

Table 2. Confusion matrix for the analyzed ANN—version 1: 29 (96.7%) tested compounds were correctly classified and 1 (3.3%) was misclassified.

	Predicted JWH	Predicted Non-JWH Cannabinoids	Predicted Others
Real JWH Cannabinoids	10	0	0
Real non-JWH Cannabinoids	0	7	0
Real Others	0	1	12

Table 3. Confusion matrix for the analyzed ANN—version 2: 27 (90.0%) tested compounds were correctly classified and 3 (10.0%) were misclassified.

	Predicted JWH	Predicted Non-JWH Cannabinoids	Predicted Others
Real JWH Cannabinoids	8	0	0
Real non-JWH Cannabinoids	0	9	1
Real Others	0	2	10

Table 4. Confusion matrix for the analyzed ANN—version 3: 26 (86.7%) tested compounds were correctly classified and 4 (13.3%) were misclassified.

	Predicted JWH	Predicted Non-JWH Cannabinoids	Predicted Others
Real JWH Cannabinoids	8	0	0
Real Non-JWH Cannabinoids	0	9	1
Real Others	0	3	9

Table 5. Confusion matrix for the analyzed ANN—version 4: 27 (90.0%) tested compounds were correctly classified and 3 (10.0%) were misclassified.

	Predicted JWH	Predicted Non-JWH Cannabinoids	Predicted Others
Real JWH Cannabinoids	8	0	0
Real non-JWH Cannabinoids	0	10	0
Real Others	0	3	9

Table 6. Data errors for the analyzed ANNs.

		Root Mean Squared Error		
Number of Nodes in the Hidden Layer	**ANN**	**Training Set**	**Selection Set**	**Testing Set**
1	Version 3	0.187558	0.276155	0.284885
3	Version 2	0.180618	0.276762	0.284408
4	Version 1	0.178430	0.241047	0.213058
6	Version 4	0.173380	0.268222	0.280003

In order to test and compare the performances of the analyzed ANNs, we used the weighted average derived from the confusion matrix, i.e., the accuracy, the recall and the F1 score (see Table 7) [14].

Table 7. Classification metrics for the target variables of each analyzed ANNs.

		Weighted Average		
Number of Nodes in the Hidden Layer	**ANN**	**Accuracy**	**Recall**	**F1 Score**
1	Version 3	0.876667	0.866667	0.866667
3	Version 2	0.903030	0.900000	0.900207
4	Version 1	0.970833	0.966667	0.967111
6	Version 4	0.923077	0.900000	0.899379

5. Conclusions

In terms of the system performance, the results obtained for the four ANNs designed to recognize the class identity of JWH synthetic cannabinoids lead to the following conclusions:

1. Accuracy [(true positives + true negatives)/total instances]:

In comparison with the accuracy (93.3%) obtained for the initial ANN model presented in a previous article, the ANN—amended version 1—generated a higher score (96.7%), while the other three ANNs generated a lower score (86.7% for the amended version 3 and 90.0% for the amended versions 2 and 4).

2. Sensitivity, or true positive rate (true positives/positive instances):

All the ANN architectures (the initial ANN and its four amended versions), had an exceptional sensitivity in detecting JWH synthetic cannabinoids (class I); all the tested JWHs were recognized as such without exception, the rate of true positives (TP) being 100% in all cases.

3. Specificity, or true negative rate (specificity = true negative/negative instances):

Compared to the specificity (90.0%) of the initial ANN model, one of the four new architectures recorded a higher score (95.0% for the amended version 1); the other three were characterized by a lower specificity (81.8% for amended version 3, 86.3% for amended version 2 and 86.3% for amended version 4).

4. Error rate [accuracy = (false positives + false negatives)/total instances]:

While the initial ANN model was characterized by an error rate of 6.6%, its amended version 1 recorded a better (lower) error rate (3.3%), while the other three ANNs recorded a higher error rate (10.0% for amended version 2, 3 and 4).

Regarding the goodness-of-fit analysis, the best results were recorded for the ANN—amended version 1 (99.97%), followed by the amended version 2 (99.94%), amended version 3 (98.41%) and amended version 4 (99.04%). Hence, we may conclude that the best performing ANN architecture was the one that included the following elements:

- Activation function: hyperbolic tangent, Softmax;
- Loss index: normalized squared error;
- Regularization: L2;
- Neuron selection: growing neurons;
- Inputs selection: growing inputs;
- Optimization algorithm: Adaptive Moment Estimation.

The results indicate that very good classification rates were obtained although the data set was complex. We intend to continue this study by applying various metaheuristic algorithms on these data sets and compare the results.

Author Contributions: Conceptualization, C.M.B. and A.C.B.; methodology, M.P.; software, A.C.B.; validation, M.P.; formal analysis, C.M.B.; investigation, C.M.B.; resources, A.C.B.; writing—original draft preparation, A.C.B.; writing—review and editing, C.M.B.; supervision, M.P.; project administration, M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: An appreciation of the "Wiley Online Library" and the data science platform and machine learning "Neural Designer", important and useful tools, used in the construction of system architectures presented in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Benitez, J.M.; Castro, J.L.; Requena, I. Are artificial neural networks black boxes? *IEEE Trans. Neural. Netw.* **1997**, *8*, 1156–1164. [CrossRef] [PubMed]
2. Doulamis, A.D.; Doulamis, N.D.; Kollias, S.D. On-line retrainable neural networks: Improving the performance of neural networks in image analysis problems. *IEEE Trans. Neural. Netw.* **2000**, *11*, 137–155. [CrossRef] [PubMed]
3. Yeung, D.S.; Cloete, I.; Shi, D.; Ng, W.W.Y. *Sensitivity Analysis for Neural Networks*; Springer: Berlin/Heidelberg Germany, 2010.
4. Montano, J.J.; Palmer, A. Numeric sensitivity analysis applied to feedforward neural networks. *Neural. Comput. Appl.* **2003**, *12*, 119–125. [CrossRef]

5. Burlacu, C.M.; Burlacu, A.C.; Praisler, M. Physico-chemical analysis, systematic benchmarking, and toxicological aspects of the JWH aminoalkylindole class-derived synthetic JWH cannabinoids. *Ann. Univ. Dunarea Jos Galati Fascicle II Math. Phys. Theor. Mech.* **2021**, *44*, 34–45. [CrossRef]
6. Todeschini, R.; Consonni, V. *Handbook of Molecular Descriptors*; Wiley: Weinheim, Germany, 2000; pp. 1–667.
7. Burlacu, C.M.; Gosav, S.; Burlacu, B.A.; Praisler, M. Convolutional Neural Network detecting synthetic cannabinoids. In Proceedings of the 2021 International Conference on e-Health and Bioengineering (EHB), Iasi, Romania, 18–19 November 2021; pp. 1–4.
8. Egger, J.; Pepe, A.; Gsaxner, C.; Jin, Y.; Li, J.; Kern, R. Deep learning-a first meta-survey of selected reviews across scientific disciplines, their commonalities, challenges and research impact. *PeerJ Comput. Sci.* **2021**, *7*, e773. [CrossRef] [PubMed]
9. Gupta, R.; Srivastava, D.; Sahu, M.; Tiwari, S.; Ambasta, R.K.; Kumar, P. Artificial intelligence to deep learning: Machine intelligence approach for drug discovery. *Mol. Divers.* **2021**, *25*, 1315–1360. [CrossRef] [PubMed]
10. Jing, Y.; Bian, Y.; Hu, Z.; Wang, L.; Xie, X.Q.S. Deep learning for drug design: An artificial intelligence paradigm for drug discovery in the big data era. *AAPS J.* **2018**, *20*, 1–10.
11. Smith, S.L.; Dherin, B.; Barrett, D.G.T.; De, S. On the origin of implicit regularization in stochastic gradient descent. *arXiv* **2021**, arXiv:2101.12176.
12. Wang, M.; Xu, X.; Yan, Z.; Wang, H. An online optimization method for extracting parameters of multi-parameter PV module model based on adaptive Levenberg-Marquardt algorithm. *Energy Convers. Manag.* **2021**, *245*, 114611. [CrossRef]
13. Singarimbun, R.N.; Nababan, E.B.; Sitompul, O.S. Adaptive moment estimation to minimize square error in backpropagation algorithm. In Proceedings of the 2019 International Conference of Computer Science and Information Technology (ICoSNIKOM), Medan, Indonesia, 28–29 November 2019; pp. 1–7.
14. Powers, D.M.W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2011**, arXiv:2010.16061.