

Effective use of neural networks simulators for image recognition and handwriting

E. Zdravkova

Faculty of Mathematics and Informatics, University of Shumen “Episkop Konstantin Preslavsky”
Corresponding author. E-mail: e.zdravkova@shu.bg

Paper received 04.12.16; Accepted for publication 10.12.16.

Abstract. This article is devoted to study the possibility of constructing neural networks using simulators for neuronal networks - NeuroPh and Joone, to recognize images and handwriting. The simulator NeuroPh has specialized interface for building this type of neural networks in a standardized and relatively lightweight process. In the simulator Joone the process is not specific as in NeuroPh. The difference is in the type of created input files (images, audio - files, text) for the simulator. The article shows and discusses the results obtained study models of neural networks implemented by simulators. Analysis is under their effectiveness, their ability to recognize and classification of objects of the same type.

Keywords: simulator, artificial neural network (ANN), image, handwriting, Backpropagation -algorithm with back propagation error

Introduction. There are various methods and algorithms for modeling artificial neural networks [1,2,3], but simulators allow for maximum closer to real applications. Neural networks are widely used in artificial intelligence systems in various areas [4]. Pattern recognition is a typical application and study their characteristics by modeling simulators have practical significance to improve their structural architecture. The simulators NeuroPh and Joone have a number of advantages as a convenient interface and tools to easily build models of various types of neural networks [5,6]. They are available free environments with GNU Lesser General Public License (LGPL) and easy to absorb. Self-learning with neural networks is one reliable method for approximating integer, continuous and target vector functions. For certain types of tasks, such as learning to interpret complex real data from sensors, artificial neural networks (ANN) are among the most effective currently known algorithms. For example, the algorithm BACKPROPAGATION is proven to be very successful in solving real-time problems such as learning to recognize handwriting, learning to recognize speech and learning to recognize faces.

Methodology. At the beginning we will realize a neural network for image recognition using the NeuroPh simulator. For this purpose, we add a new file to a new project or to an existing one on the simulator. This is done from the menu File -> New File. In the dialog box that appears we set the following parameters: in the field Categories we choose Neuroph, and in the field File Types we choose Image Recognition. In the next dialog box we choose the images, that the neural network must learn to recognize. Here we will introduce four black and white images of animals –dog, cat, bird and rabbit.



Fig. 1. Black and white images of four animals, that the neural network will be trained to recognize

After that, we point out images, that the neural network must not recognize. According to the standard, we introduce full of color images. In this case we introduce images of three colors – red, green and blue.



Fig. 2. Colors that neural network should not recognize

On the next dialog box we choose the parameters of the training set. In the last dialog box we have to set the parameters of the neural network – layers and number of neurons in them. They are separated with space. We set the neural network with two hidden layers, with 10 and 5 neurons respectively. The resulting neural network is shown in Figure 3.

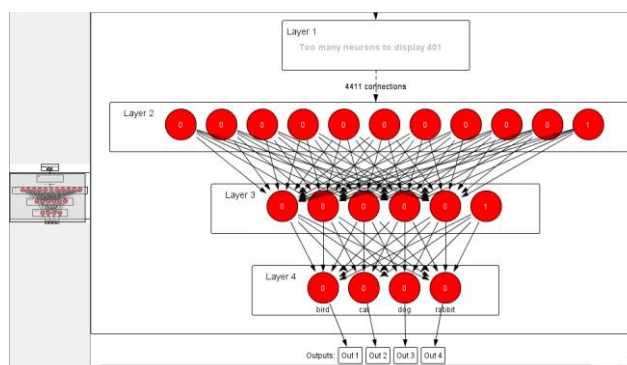


Fig 3. Neural network for image recognition on the NeuroPh simulator

We start a test of the neural network before the training by setting any of the images. The results are the following:

Table 1. Results before training of the neural network

Image	Result
rabbit	0,5041
bird	0,4835
cat	0,4791
dog	0,4583

Next, we have to train the neural network with the teaching set `setDataImage_R`, that NeuroPh automatically generates with the creating the neural network and pointing the images. It should be noted that the number of neurons in the input layer are also automatically generated in this simulator. Initially we set training with the following parameters:

MaxError: 0.01
 Learningrate: 0.2
 Momentum: 0.7

The result of the average error of the neural network is the following:

realtimeerrorgraph: 0.4839214496564955

The results for the images are now the following:

Table 2. Results after the first training

Image	Result
rabbit	0,4811
bird	0,3989
cat	0,379
dog	0,3194

We run one image to also test the average error. The results of the four outputs are:

0.39886636055003950.3789529685439983
 0.319416035985367 0.4810967656261946
 0.36698451285161690.3573778334082419
 0.2987064918439039 0.5099213203378068
 0.3726749794985280.3639529507938539
 0.3042067217592674 0.5036297705085665
 0.399918218293542460.37799134936167517
 0.3202091909135918 0.4829880264359195

We run another test with the neural network. This time we start training with 10000 times and again using the same values given above. The result for the average error of the neural network is now the following:

realtimeerrorgraph:0.4513606436201545

The results for the images after this training are as follows:

Table 3 . Results after the second training of the neural network

Image	Result
rabbit	0,4405
bird	0,2178
cat	0,1455
dog	0,0952

We run a test again by an image for the error on each output. The results are:

0.217832755424704870.14552383612806305
 0.09523453425105093 0.4405222562596394
 0.070503342071209910.10168257648621529
 0.05219827313441213 0.6907314827380383
 0.066446852736996090.10785815009604982
 0.05281880214748554 0.6901361333308271
 0.219077850206344740.15568442127093485
 0.10986737737315685 0.4941143270524672

The next step is to realize the neural network for handwriting recognition. For this purpose we add a new file again, usingFile -> New File. In the dialog window that appears, we choose Neuroph in the field Categories, and in the field-File Types we choose Hand Written Character Recognition. In the next dialog window we enter the letters that the neural network must learn to recognize. This happens as they are at first written by hand in the field Draw Character, and then saving them with the button Save. On the dialog box that appears we enter from the keyboard the equivalent of the handwritten letter. After that the entered handwritten symbols are transferred to the field Training Set Characters. Initially, this study will introduce recognition of two handwritten symbols – the letters a and b. In the last dialog box we have to set the parameters for the neural network –

layers and number of neurons in them. They are separated with space. We set the neural network with two hidden layers with 10 and 3 neurons respectively. The resulting neural network is shown in Figure 4.

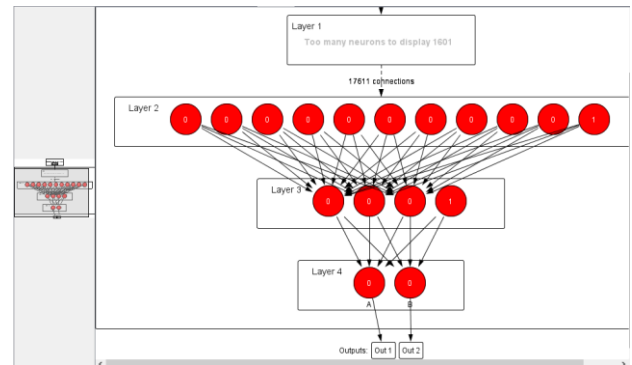


Fig. 4. Neural network for handwriting recognitions using the NeuroPh simulator

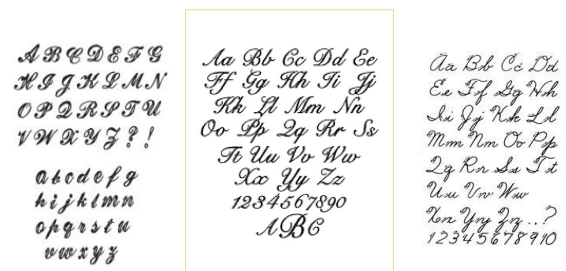


Fig. 5. Images for training and testing of the neural network for handwriting

The results before the training of the neural network are as follows:

Table 4. Results before the training of the neural network

A	B
0,7277	0,3003

When running a test for the average error in the network, the results of the two outputs are:

0.71936469809176760.30875988662128767
 0.73022752587504680.29502008298321364

Next, we have to train the neural network with the training set NewLettersTrainingSet. It is again automatically generated by the NeuroPh simulator during the creation of the neural network and defining the symbols that it will recognize. The number of neurons in the input layer is also automatically generated by the simulator. We start training with the following parameters:

MaxError: 0.01
 Learningrate: 0.2
 Momentum: 0.7

The result for average error of the neural network after the training is the following:

realtimeerrorgraph: 0.2737016103164107

The results for recognizing the two symbols are as follows:

Table 5. Results after the training of the neural network

A	B
0,886	0,2098

Now we run a test for the average error and the results of both outputs are as follows:

0.85981925932432930.25057378150350246
0.87691534136070450.1827334429059068

If we conclude from the data received, it is noticed that the time for training of the neural network for symbol recognition is significantly less.

The next thing we need to do for the study is to create two neural network in the Joone simulator too. We start with neural network for image recognition. The neural network will recognize images of animals and here we use the same four sample images that we used in the previous example (Figure 1). We start Joone and create several layers of the neural network with sigmoidal connections. The first layer is the input layer with 401 neurons. We create it the same way as the neural network in the NeuroPh simulator. Then two hidden layers as follows again with sigmoidal connections. The first hidden layer is with 10 neurons, and the second – with 5. The last layer of the neural network is the output layer that actually has 4 neurons..

A test set is also present in the neural network. It is passed as an input image file, also training set, connected with the so called Trainer. Finally the result is obtained in and output file Output. Here it is very significant to use an image file for both input and the output set. We pass the same parameters for training the neural network, as the parameters used in the Joone simulator. We pass for training the bird image from the sample images that the neural network must learn to recognize. It turns out, however, that the simulator can't train the neural network and can't start a test set, because the number of input neurons is not compatible with the given set. We change the input neurons according to the file size - 144 x 90. This way we create a neural network with 12960 neurons in the input layer. The resultant network is shown in Figure 5.

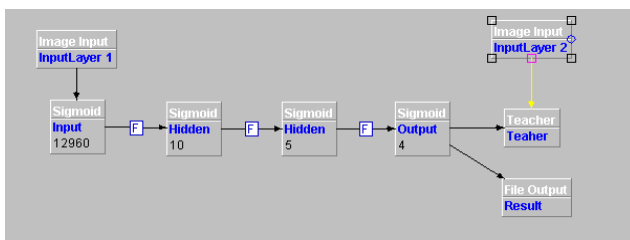


Fig. 5. Neural network for image recognition in Joone

Next is creating the neural network for handwriting recognition in Joone simulator. Here we have the same number of layers in the network. In the input layer we have 1601 neurons, in the first hidden layer we have 10 neurons, and in the second hidden layer - 3. In the output layer in this case we have 2 neurons and here this is because we want to recognize two symbols "a" and "b".

In the neural network are present again a test set, that is passed as an input image file, and also a training set, connected with the so called Trainer. Finally the result is obtained in the output file Output.

For the training of the neural network we set the same parameters we used with the Joone simulator. We use as input for training one of the images with handwriting, shown in Figure 7. It turns out that once again the simulator can't start the training and it can't pass the training set to the network, because the number of input neurons is not

suitable. In this case we change the number of neurons again to match the file size. Here the image file with handwriting has the size of 300 x 300, so we change the number of input neurons to 90000. The resulting neural network is shown in Figure 6.

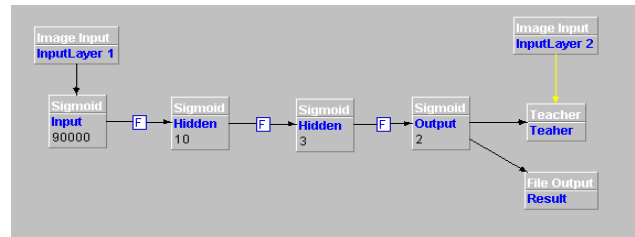


Fig. 6. Neural network for handwriting in Joone

Both neural networks, created by the Joone simulator don't give good results during the tests. This shows that there clearly are differences in creation, testing and using of neural networks in the two simulators - Joone and NeuroPh.

Conclusions

After testing of the two simulators for neural networks - Joone and NeuroPh, there were noticed some peculiarities in creation, testing and training of the neural networks.

It is noticeable that the neural networks for handwriting are training a lot faster than the neural networks for image recognition. Here, however, the process is also determined by the neural network itself and the number of neurons in it.

The neural network for image recognition in the NeuroPh simulator has 401 neurons in the input layer and in the Joone simulator we created neural network with 12960 neurons. We explained above, that this is due to the size of the input image file, which must be taken into account by the input layer of the neural network in this simulator. The situation with the neural network for handwriting recognition is similar. In the NeuroPh simulator the neural network for handwriting has 1601 neurons in the input layer, and the created with the Joone simulator neural network is with 90000 neurons. So there are differences in the very structure and architecture of the neural networks. Parameters of the neural network that we pass in one simulator can't always be applicable to the other.

In the study we noticed the quite simple interface of the NeuroPh simulator for creating neural networks for recognition. In this simulator there is a special module and corresponding file type for this kind of neural networks. The number of the neurons in the input layer, as well as the size of the training set in this simulator is determined automatically during the input of images that this neural network must recognize. In the Joone simulator, the neural networks are created in a standard way and there is no special interface for neural networks for recognition.

It can be summarized that the NeuroPh simulator can be used by novice neural networks programmers, who just started using neural networks for recognition and explore their features. The Joone simulator can be used by programmers who are advanced in creating neural networks for recognition and are already familiar with their characteristics.

REFERENCES

1. Nachev, A., B. Stoyanov, MLP and Default ARTMAP Neural Networks for Business Classification, Intelligent Decision Making System, Proceedings of the 4th International ISKE2009 Conference, World Scientific Series

- on Computer Engineering and Information Science 2, 179-185. http://e-proceedings.worldscinet.com/9789814295062/9789814295062_0028.html
- Nachev, A., Hill, S., Barry, C., Stoyanov, B., Fuzzy, distributed, instance counting, and default artmap neural networks for financial diagnosis, International Journal of Information Technology & Decision Making (IJITDM), Vol. 9, No. 6(2010) pp. 959-978, DOI: 10.1142/S0219622010004111, Impact Factor 2010 = 3.139. <http://www.worldscinet.com/ijitdm/09/0906/S0219622010004111.html>
 - Nachev, A., Hogan, M., Stoyanov, B., Cascade-Correlation Neural Networks for Breast Cancer Diagnosis, Proceedings of the 2011 International Conference on Artificial Intelligence, ICAI 2011, July 18-21, 2011, Las Vegas, USA, 2 Volumes 2011, IC-AI 2011: 475-480. <http://world-comp.org/p2011/ICA4110.pdf>
 - Nenkov Nayden, Petrova Mariana, Dyachenko Yuriy, Intelligence technologies in management and administration of justice, 3rd International Multidisciplinary Scientific Conferences SOCIAL SCIENCES & ARTS SGEM2016, POLITICAL SCIENCES, LAW, FINANCE, ECONOMICS & TOURISM, Conference proceedings, Volume V. pp. 385-392, 2016. ISBN 978-619-7105-76-6, ISSN 2367-5659, DOI 10.5593/segemsocial2016B25.
 - Zdravkova, Elitsa., Nenkov, N.V., Comparative analysis of simulators for neural networks Joone and NeuroPh, (On-line: www.cmnt.lv), ISSN 1407-5806, ISSN 1407-5814, Transport and Telecommunication Institute, Riga, Latvia, 2016, 20(1) 16-20.
 - Zdravkova, Elitsa., Nenkov, N.V., Research of neural network simulators through two training data sets, (On-line: www.cmnt.lv), ISSN 1407-5806, ISSN 1407-5814, Transport and Telecommunication Institute, Riga, Latvia, 2016 20(1) 12-15.

ЛИТЕРАТУРА

- Начев, А. Б. Стоянова, MLP и по умолчанию ArtMap нейронных сетей для бизнеса классификации, интеллектуальной системы принятия решений, Труды 4-й InternationalISKE2009 конференции, Всемирной научной серии по компьютерной инженерии и информатики 2, 179-185. http://e-proceedings.worldscinet.com/9789814295062/9789814295062_0028.html
- Начев, А. Хилл, С., Барри, С., Стоянов, Б., Нечеткие, распределенных, подсчет экземпляров и ArtMap по умолчанию нейронных сетей для финансовой диагностики, Международный журнал информационных технологий и принятия решений (IJITDM), Vol , 9, № 6 (2010) С. 959-978, DOI: 10,1142 / S0219622010004111, Impact Factor 2010 = 3,139. <http://www.worldscinet.com/ijitdm/09/0906/S0219622010004111.html>
- Начев А., Хоган, М., Стоянов, В., Каскад-Корреляция нейронных сетей для диагностики рака молочной железы, Труды 2011 Международная конференция по искусственному интеллекту, ИПБИ 2011, 18-21 июля 2011 года, Лас-Вегас, США, 2-х томах 2011, IC-AI 2011: 475-480. <http://world-comp.org/p2011/ICA4110.pdf>
- Ненков Найден, Петрова Мариана, Дьяченко Юрий, технологии разведки в области управления и правосудия, 3-й Международный многопрофильной Научные конференции СОЦИАЛЬНЫЕ НАУКИ И ИСКУССТВА SGEM2016, политология, право, финансы, экономика и туризм, Материалы конференции, Том V. стр. 385-392, 2016 г. ISBN 978-619-7105-76-6, ISSN 2367-5659, DOI 10.5593 / segemsocial2016B25.
- Здравкова, Элица, Ненков Н.В., Сравнительный анализ тренажеров для нейронных сетей Joone и NeuroPh, (онлайн: www.cmnt.lv), ISSN 1407-5806, ISSN 1407-5814, Институт транспорта и связи, Рига, Латвия, 2016, 20 (1) 16-20.
- Здравкова, Элица, Nenkov Н.В., Исследование нейронных сетей тренажерах через два набора обучающих данных (Он-линии: www.cmnt.lv), ISSN 1407-5806, ISSN 1407-5814, Transport and telecommunication institute electrosвязи, Рига, Латвия, 2016 20 (1) 12-15.

Эффективное использование нейронных сетей тренажеров для распознавания образов и рукописного ввода

Э. Здравкова

Аннотация. Данная статья посвящена изучению возможности построения нейронных сетей с использованием имитаторов для нейронных сетей - NeuroPh и Joone, распознавать образы и почерк. Тренажер NeuroPh имеет специализированный интерфейс для создания такого типа нейронных сетей в стандартизированном и относительно легкий процесс. В тренажере Joone процес не является специфическим, как и в NeuroPh. Разница заключается в типе созданных исходных файлов (изображений, аудио-файлы, текст) для тренажера. В статье показано, и обсуждаются полученные результаты исследования модели нейронных сетей, реализованных тренажерах. Анализ находится под их эффективности, их способность распознавать и классификации объектов того же типа.

Ключевые слова: тренажер, искусственные нейронные сети (ИНС), образ, почерк, -алгоритм обратного распространения ошибки с задней распространения