Framework for the Interactive Learning of Artificial Neural Networks

Matúš Užák and Rudolf Jakša

Department of Cybernetics and Artificial Intelligence
Technical University of Košice, Slovakia
uzak@neuron.tuke.sk, jaksa@neuron.tuke.sk

Abstract. We propose framework for interactive learning of artificial neural networks. In this paper we study interaction during training of visualizable supervised tasks. If activity of hidden node in network is visualized similar way as are network outputs, human observer might deduce the effect of this particular node on the resulting output. We allow human to interfere with the learning process of network, thus he or she can improve the learning performance by incorporating his or her lifelong experience. This interaction is similar to the process of teaching children, where teacher observes their responses to questions and guides the process of learning. Several methods of interaction with neural network training are described and demonstrated in the paper.

1 Introduction

The process of learning of artificial neural network can be visualized, observed, and interactively guided by a human observer. Traditionally, learning of artificial neural networks is treated as adaptation of a black box, although works focused on visualization what happens inside the box can be found dating back to beginnings of the field. On the other side, establishment of the Interactive Evolutionary Computation (IEC) domain brings into forefront the idea of interactive intervention into algorithm by a human observer. This idea of interactive guidance of algorithm has to be explored in the neural networks domain too.

Combination of the learning algorithm, which searches a weight space of the network, and a human observer, which gains an overview over the behavior of algorithm and is able to guide this algorithm, may bring some new possibilities into the field of neural networks. Experience of a human might be usable for the algorithm to escape from the local minima trap. Ability to guide the learning might bring new tasks for neural networks, not strictly defined by a training data set. Basic method for incorporation of a human observer into learning process is the visualization. This is used in IEC domain and also studied in the past in neural networks area. Easiest tasks to visualize are these which are defined in two-dimensional space and naturally have a visual character, although three-dimensional, motion video or these with audio character may by visualized or another way presented to a human observer too. Survey of tasks and also methods studied in IEC area is provided in [1].

S. Kollias et al. (Eds.): ICANN 2006, Part I, LNCS 4131, pp. 103–112, 2006.

The aim of most neural network visualization techniques is to help to understand what neural networks really do. The essence of each technique lies in visualization of some object common to all neural networks: topology, response to processed data, internal mappings. Although in general, the target of all techniques is to visualize internal mappings, different paradigms are used. This is mostly due to fact, that in different applications, different aspect of network performance are studied.

First visualization techniques used were Hinton and Bond diagrams described in [2]. These techniques help to analyze the input units importance through the magnitude of weights connecting them with hidden units. However, the analysis of internal mapping is reduced to analysis of weights. These methods display the topology of network. Craven in [3] also accompanied them with trajectory diagrams. Craven's visualization tool called *Lascaux*, did visualize many objects relevant only for single training vectors: the activation signal, and error signal propagated through the network, but its essence was in modification of Bond diagrams implementation. This modification is later referred to as the network interpretation diagram (NID) used by Olden [4], similar modification implemented in three-dimensional space was done by Edlund [5]. Olden used Garson's diagrams to visualize the input nodes importance to the outputs and the sensitivity analysis as a method to comprehend the inner mappings.

Important is also the work of Streeter [6] where is described a visualization tool that provides opportunities for interaction in the learning process. Learning is realized through the error backpropagation algorithm or using Evolutionary Strategy (ES). Human observer is able to manually adjust weights and their learning rate in the backpropagation case, or mutation rate in the ES case. They used a modification of NID, and a modification of Hinton diagrams to visualize as many networks in the evolution process as possible.

Recently were introduced methods which analyze the performance of network by observing its reaction to processed data. Interesting projection techniques have been described by Duch, [7][8], that should help to analyze the network performance through visual interpretation of decision borders. Tzeng [9] introduced a modification of NID with built in representation of modified Carson's method that analyzed hidden units importance. They analyze the network in data-driven approach.

Our own work is focused to studying reactions of single units to the testing set which represents the task. As weights are a part of the unit they are connected to, the visualized information is reduced compare to methods that visualize weights. This method is described in [3] as a Response Function Visualization. Similar to hyperplane plots [3], this method is constrained to two-dimensional tasks, but it is obvious that combined with methods proposed by Hinton, Tzeng and Duch, this limitation can by surpassed.

Generally, the challenge is to combine the best from mentioned methods to provide the human observer with only the most valuable information about the learning process, and allow him or her to maintain full concentration to interactive interventions.

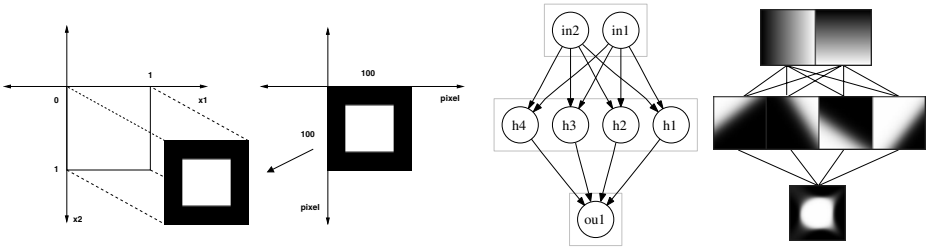

Fig. 1. (*Left*) The image with dimensions 100x100 pixels mapped to space of classification task. (*Right*) Example of the response of network units to the function signal propagating through the network. Images were created by presenting full set of testing vectors from the space of classification task. We classify the vectors inside and outside of given square.

2 Visualization of Learning of Neural Network

We will describe the design of visualization method of learning process using the multilayer perceptron concept. The multilayer perceptron consists of a layer of input units, of one or more hidden layers and an output layer. Computational units of the network are in output layer and in hidden layers. Proper visualization of behavior of these units during the learning should allow to reason about the learning process itself.

We will visualize the learning of a classification task. The space, where the classification takes place is projected into two-dimensional (2D) space for visualization. Training data are randomly chosen from samples generated from classification task space. If the classification task itself is defined in the 2D space, the 2D image representing classification results can be easily generated as a direct mapping of this space into image space. Figure 1 clarifies the procedure.

The goal of the visualization is to capture responses of individual neurons to the function signal during the whole testing phase. The function signal is an input signal – stimulus, that comes in the input end of the network, propagates forward (neuron by neuron) through the network, and emerges at the output end of the network as an output signal [10]. The signal itself is a set of vectors from defined space. Testing phase is a presentation of defined set of input vectors to the network and propagation of function signals through the network. In addition to visualization of responses of active neurons – computational units of network, also input units – entrances of input signal can be visualized to extend the observer's view. Thus the function signal can be completely observed from its entering point to the output point in the network. This method is also referred as the response function visualization [3]. With visualization the observer can track the reactions of individual neurons to the propagating signals. Visualization of input units provides the information about the intensity of the signals which enter the network.

The visualized area of classification space might be extended to cover also places which are far from the training data. By observing behavior of network in these distant areas an observer may reason about extrapolation/generalization qualities of network. Such acquired knowledge should help to uncover possible problems with using the network in situations where it can be confronted with the samples from outside the area of maximum classification confidence [11]. See Fig.2 for an example of extrapolation test.

Fig. 2. (*Left*) The network was trained to classify vectors inside the framed square in the middle of image. The another big square is an extrapolation artifact and is related to poor classification performance on the upper left corner of original square. The training data are only from the framed area. (*Right*) Extrapolation artifacts on spiral classification.

3 Interactive Intervention into Learning

Human observer of the learning process of neural network may be allowed to interfere with this learning. Such intervention should allow for incorporation of his or her lifelong experience into the learning process. We will study several alternatives of interactive intervention into learning:

- amplification of outputs of neurons,
- amplification of inputs of network,
- manual adjustment of bias of neuron,
- adjustment of individual learning parameters of neurons,
- reinitialization of individual neurons.

Some of these interventions require small modifications of learning algorithm or modification of structure of neural network. We will use error backpropagation algorithm for the learning of network. Consider multilayer perceptron neural network with neuron activations x_i, link weights w_{ij} which links the j-th neuron into i-th neuron, biases (thresholds) θ_i, and neuron activation functions $f_i(in_i)$, described by (1). The in_i is input into i-th neuron and M is the number of links connecting into i-th neuron. Gradient based error minimizing adaptation of weights follows (2), with the γ learning rate constant, and δ_i error signal. The $f'(in_i)$ is the derivative of activation function $f(in_i)$. Error signal δ_i for output

neurons can be computed using (3a), but for neurons in hidden layer the (3b) should be used instead. The N_h is number of links coming from i-th neuron and h is index of these links and corresponding neurons.

$$x_i = f_i(in_i), \quad in_i = \sum_{j=1}^{M} w_{ij} x_j + \theta_i \tag{1}$$

$$\Delta w_{ij} = \gamma \delta_i x_j \tag{2}$$

$$\delta_i = (ev_i - x_i) f'(in_i), \quad \delta_i = f'(in_i) \sum_{h=1}^{N_h} \delta_h w_{hi} \tag{3}$$

Rule (3b) is the error backpropagation rule, it defines the backward propagation of error through network. Rule (2) defines weight changes for minimization of this error, and (3a) defines initial error signal on the network output. The error backpropagation algorithm is defined by rules (1), (2), and (3).

3.1 Amplification of Outputs of Neurons and Inputs of Network

The influence of individual neuron on the function of whole network is weighted by weights on links connected to this neuron. We add another weight onto output of every neuron to further control influence of individual neurons. This weight is manually adjusted by a human observer during the learning. Let's call this weight a master weight (mw). Equation (1) of network description has to be changed to (4), so the activations of neurons are multiplied with this mw. Further, backpropagation algorithm rules (3) have to be modified into Neural network usually converges during the learning to a certain stable point, where the weights change only slightly. During this convergence, sudden amplification of outputs of neurons by a human observer may have an interesting effect, mainly in stability disruption. (5).

$$x_i = f_i(in_i) mw_i \tag{4}$$

$$\delta_i = (ev_i - x_i) f'(in_i) mw_i, \quad \delta_i = f'(in_i) mw_i \sum_{h=1}^{N_h} \delta_h w_{hi} \tag{5}$$

$$in_i = in_i^{orig} ss_i \tag{6}$$

Similarly to amplification of outputs of neurons, inputs of the network might be amplified too. Human observer then gains ability to control the amplitude of signals which enter the network. Optimal amplification for the learning of given task then can be searched for. Let's call this amplification parameter a sensoric strength (ss). We must introduce a new equation (6) to describe this modified amplified input signal, where in_i^{orig} is original input signal without any amplification. The input signal after its amplification or attenuation is further propagated through the network. Learning algorithm will not be affected by the amplification but it is important to consider the amplified signal when computing the weight changes for the neurons of the first hidden layer.

3.2 Manual Adjustment of Bias of Neuron

Manual adjustment of bias value of neuron allows for balancing the output of given neuron – shifting the output of neuron to bigger or lower values. Bias or threshold is one special weight of every neuron in network. It is used to balance the activation of a neuron, responding to its inputs. Let's call additional bias weighting parameter a master threshold (mt). Equation (1) will be changed into (7), and the (2) when used for a bias adaptation must be modified too, into (8).

$$in_i = \sum_{j=1}^{M} w_{ij} x_j + \theta_i mt_i \tag{7}$$

$$\Delta\theta_i = \gamma \delta_i mt_i \tag{8}$$

3.3 Adjustment of Individual Learning Parameters of Neurons

Manual adjustment of individual learning rate parameters of backpropagation algorithm for individual neurons allows for control of adaptation rate of neurons. This allows a human observer to freeze well responding neurons, and set up more aggressive learning rate for these with not so good response. In algorithm we must introduce individual learning rate parameters γ_i instead of fixed learning rate γ in standard error backpropagation algorithm. Equation (2) is then changed into (9).

$$\Delta w_{ij} = \gamma_i \delta_i x_j \tag{9}$$

3.4 Reinitialization of Individual Neurons

More radical form of dealing with not well responding neurons is their reinitialization. Reinitialization of particular neuron is done by setting of weights on its links into random values from interval used for initialization of neural network. Reinitialization of important neuron can damage also the behavior of neurons connected to it, however reinitialization of not well responding neuron should be not so damaging as there is a chance, that other neurons are not tightly connected to this particular poorly behaving neuron. Useful practice is to lower learning rate to all well behaving neurons prior such reinitializations, to mitigate possible damage. Reinitialization of not well responding neurons is quick and easy to do for a human observer. It can be used to deal with neurons in the saturation stage, or to explore the parameter space of neural network during learning by observing convergence of reinitialized neurons.

The spectrum of intervention methods should allow a human observer of learning of neural network to exploit a maximum possibilities of such interaction. However the bigger this spectrum is, the more time for learning how to use them is necessary.

4 Experiments

We will study easily visualizable classification tasks. We will run them through visualized interactive learning with multilayer perceptron networks and error backpropagation algorithm and describe our experience with this interactive learning. Visualization method described in Sect.2 will be used for visualization and methods described in Sect.3 will be used for interaction.

Tasks of classification of two-dimensional geometrical forms of circle, spiral, square, and square frame will be studied. See Fig.3 for exact view of these forms. The classification task itself is to classify whether a given point falls inside the form or outside it. The point is defined by its coordinates.

Fig. 3. Graphical interpretation of studied classification tasks

The aim of experiments is to try to extract the know-how to enable effective solution of classification tasks and also effective usage of interactive interventions. The effectiveness is evaluated in terms of quality of acquired knowledge. The network is evaluated by observing how was the resulting knowledge built, i.e. which building blocks were formed during the learning, connected together and modified for error minimization. These building blocks are represented by visualized responses of hidden layers neurons to function signal.

4.1 Network Inputs Amplification and Reinitialization of Neurons

Low levels of the input signals which enters the network may cause learning problems. The confirmed premise for experiments was, that after increasing of levels of input signals, backpropagation learning should be able to keep the correct direction for approaching the closest minimum in error space from the start of learning, thus avoiding possible saturations. See Fig.5 for an example.

Reinitialization of neurons is the fastest correctional intervention into the learning process out of the interactive interventions described previously in the paper. The experiments showed that it is also probably the most useful interactive intervention. However, when most of the neurons were saturated, reinitializations did not have a required effect. After reinitialization, the weights did return to their original values, and the rest of saturated network did not move its weights in any direction. Actual weight vector can not escape from a local extreme when certain amount of neurons are saturated.

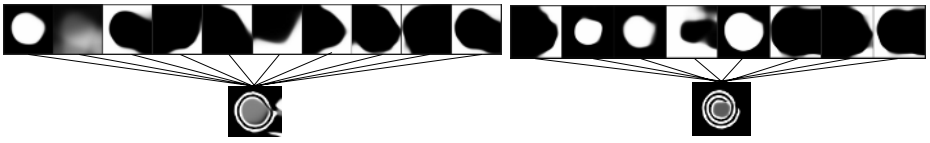

Fig. 4. A larger number of hidden layer neurons does not guarantee for qualitatively better resulting knowledge. More important is, which parts does the resulting knowledge comprise of. Two networks learning the spiral classification were trained in the same number of iterations. Only the second hidden layer is displayed on the figure.

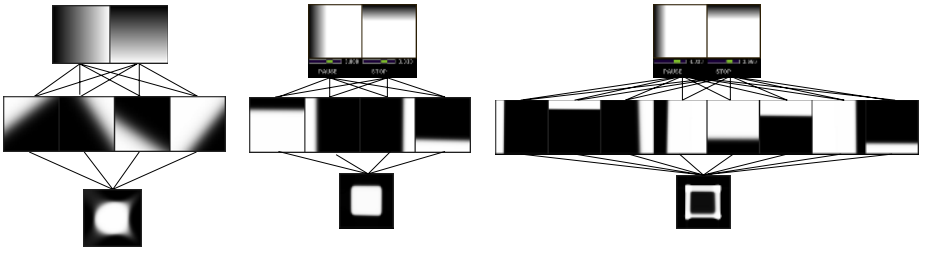

Fig. 5. (*Left*) Average best result acquired by the backpropagation learning for the square classification with a minimal topology without the input signals amplification. (*Right*) Results obtained with optimized input signal amplifications.

4.2 Adjustment of Individual Learning Parameters of Neurons

Adjustment of individual learning rate parameters of neurons has proved itself as a suitable complement for reinitialization of neurons that responded unfavorably to a function signal. It is useful to set the learning rate parameter for well responding neurons to the value close to zero. This fixes the favorable building blocks and only unfavorable blocks are further changing. It enables human observer to reduce the search-space of learning algorithm according to his visual impression. Setting the learning rate parameter to zero causes that the search will not move along the given axis anymore. Left part of Fig.6 represents an example of described procedure.

4.3 Manual Adjustment of Bias of Neuron

Our experiments were focused on question how bias adjustments affects the resulting knowledge. The interactive adjustments were incorporated during the learning, and after the learning too. The goal is to improve acquired knowledge.

Problem with this approach are quick reactions of learning algorithm to user interventions. Learning algorithm quickly balances weights of neurons to compensate interventions, and this limits usability of bias adjustments. Also reactions of other neurons are sometimes dramatic and hard to predict with this approach. However, bias adjustments did not disrupt the stability of learning behavior of the network as much as the signal amplifications. Bias adjustments

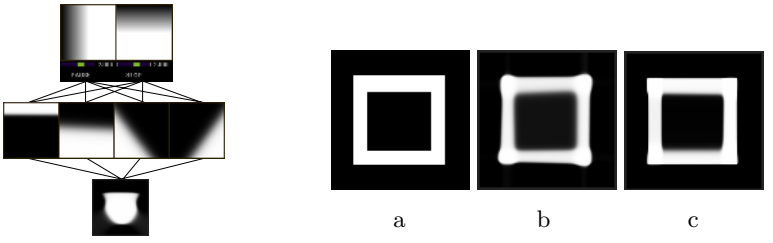

Fig. 6. (*Left*) First two hidden layer neurons respond well to the second input – sensor. To enforce influence of their behavior, their learning rate parameters have to be set to small values, while rest two neurons have to be reinitialized. (*Right*) Example of a result achievable with a proper bias adjustment and signal amplifications, (*a*) is the optimal result we want to obtain, (*b*) is result of learning prior interactive intervention, and (*c*) is almost flawless classification after interactive refinement.

proved useful after the learning was finished, to refine final behavior of network, see right part of Fig.6.

4.4 Amplification of Outputs of Neurons and Virtual Input Units

With various task types and topologies we found interactive amplifications of neurons outputs prone to damaging the knowledge of the network acquired before their application. These amplifications can easily destabilize learning and they can lead into oscillations in learning behavior. Simple amplification of outputs of neurons as described by (5) seems not well suited for interactive interventions. However, as with bias adjustments, amplification of outputs of neurons is useful after the learning is finished, it is demonstrated on the left part of Fig.7.

When solving more complicated tasks, it is reasonable to create functional links [12] feeding the network with virtual signals formed by certain functions applied on real input signals of network. Addition of these virtual inputs makes an aid for neurons on the first hidden layer. These neurons can use this added degree of freedom in the weight space for adjustment of their behavior. See right part of Fig.7 for an example.

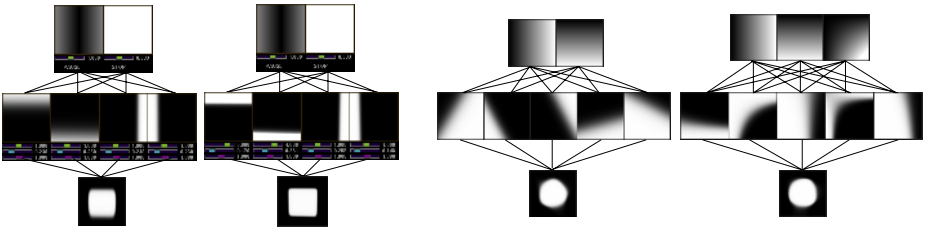

Fig. 7. (*Left*) Decision borders represented by the first two hidden neurons are blurred, which slightly deforms resulting knowledge. After the learning, signal intensities from these neurons were amplified, which qualitatively improved resulting classification. (*Right*) Several neurons from hidden layer respond to added virtual input unit, a result is a smoother decision border for circle classification.

5 Conclusion

Interactive interventions into learning of neural networks shift the focus of learning from simple error correction to a process guided by a human observer – teacher. The goal can be just acceleration of learning pace, but it can move into ability to embed human knowledge into learning process. As it is an interactive process it also allows a human observer to better understand learned behavior of a neural network by observing consequences of his or her interventions.

The framework for interactive learning presented in this paper differs from this used in more established field of interactive evolutionary computation. The main difference is requirement of expert knowledge from neural network area, which makes it a research tool. We hope in future interactive learning of neural networks can be refined enough to be used in real world applications.

Our presented experiments demonstrate usefulness of approach and show the characteristics of particular methods of intervention. Our future research is focused on neural networks with a big number of neurons, where we apply clustering methods to limit the amount of information for a visualization.

References

1. Takagi, H.: Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. Proceedings of the IEEE **89**(9) (2001) 1275–1296
2. Weichert, J., Tesauro, G.: Visualizing processes in neural networks. IBM J. Res. Develop. **35** (1991) 244+
3. Craven, M.W., Shavlik, J.W.: Visualizing learning and computation in artificial neural networks. International Journal on Artificial Intelligence Tools (1) (1991) 399–425
4. Olden, J., Jackson, D.: Illuminating the "black box": Understanding variable contributions in artificial neural networks. Ecological Modelling **154** (2002) 135–150
5. Edlund, M., Caudel, T.: Realtime visualization of the learning processes in the lapart neural architecture as it controls a simulated autonomous vehicle. Proceedings of the International Joint Conference on Neural Networks **3** (2000) 41+
6. Streeter, M., Ward, M., Alvarez, S.A.: Nvis: An interactive visualization tool for neural networks. Proceedings of SPIE Symposium on Visual Data Exploration and Analysis **4302**(8) (2001) 234–241
7. Duch, W.: Coloring black boxes: visualization of neural network decision. Proc. of International Joint Conference on Neural Networks (IJCNN) **1** (2003) 1735–1740
8. Duch, W.: Visualization of hidden node activity in neural networks: I. visualization methods. Lecture Notes in Computer Science **3070** (2004) 38–43
9. Tzeng, F.Y., Ma, K.L.: Opening the black box - data driven visualization of neural networks. Proceedings of IEEE Visualization '05 Conference (2005) 383–390
10. Haykin, S.: Neural Networks: A Comprehensive Foundation. Macmillan College Publishing Company, Inc., New York, USA (1994)
11. Užák, M.: Visualization and interaction in the process of neural network learning. Master's thesis, Technical university of Košice (2005) in Slovak.
12. Pao, Y.H.: Adaptive pattern recognition and neural networks. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1989)