Efficient Learning of Contextual Mappings by Context-Dependent Neural Nets

Piotr Ciskowski

Wroclaw University of Technology
Department of Electronics
Institute of Engineering Cybernetics
Wybrzeze Wyspianskiego 27
Wroclaw 50-370, Poland
Piotr.Ciskowski@pwr.wroc.pl

Abstract. The paper addresses the problem of using contextual information by neural nets solving problems of contextual nature. The model of a context-dependent neuron is unique in the fact that allows weights to change according to some contextual variables even after the learning process has been completed. The structures of context-dependent neural nets are outlined, the Vapnik-Chervonenkis dimension of the single context-dependent neuron and multilayer net shortly discussed, decision boundaries are analyzed and compared with the traditional nets. The main goal of the article is to present highly effective contextual training algorithms for the considered models of neural nets.

1 Introduction

The phenomena and objects in real world can rarely be considered in isolation. The parameters describing them often change according to some environmental conditions, called the context. A learning machine, dealing with such systems should take the information about the context into consideration in order to improve its performance obtained using only the primary features of the analyzed object. The paper presents a novel way of context-sensitive learning for the neural network approach. A brief discussion will be presented on how context-dependent neural nets (CD-nets) use contextual data (analysis of various contextual approaches towards machine learning may be found in [1], the definitions of contextual variables and methods of managing contextual information may be found in [2,3]).

Many algorithms have been developed for improving the training process of neural nets, their generalization abilities or optimizing the nets' architecture. However, few of them intrude the neuron's model itself in order to enrich its processing abilities. Although neural nets are proved to be able to perform any input-output mapping with the desired accuracy, provided they are complex enough and given enough training data, the need is clear for designing less complex networks and more appropriate for the problems to be solved.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3037, pp. 196–203, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The model of a context-dependent neural net, introduced in [4,5], developed in [1], and presented in this paper proves to be useful for applications of neural nets in problems showing contextual dependencies among data. It is based on the ideas of contextual learning of nonlinear mappings, presented in [6], [7], [8] and [9], and applied for tasks of robot arm control and classification of ECG signals. It is a generalization of the traditional neuron's model, however, as shown in the paper:

1. more appropriate for contextual learning,
2. of higher Vapnik-Chervonenkis dimension (better processing abilities),
3. using its abilities more efficiently than just considering the growth with the number of parameters,
4. using the Kronecker product of the vectors of primary inputs and contextual basis functions, what simplifies the standard training algorithms and allows developing highly effective contextual learning algorithms.

Generally speaking we define the **context** as *all the factors that should influence (improve) the decision making* (or other kind of information processing) performed by the learning algorithm, *but other from the data on which the decision is taken* (or which are directly processed). Contrary to the traditional models of neural nets, the contextual approach makes distinction between these two groups of features and takes contextual dependencies between them into consideration in order to improve the process of machine learning. The division of features into these two groups is a separate problem and subject of research, not covered in this paper.

2 Models of Context-Dependent Nets

2.1 Model of a Context Dependent Neuron

Unlike the traditional neural net, the CD-net does not transform the primary and contextual features equally. It groups primary and context-sensitive inputs in the primary input vector $\overline{\mathbf{X}}$, operates on them similarly as the traditional net, although making the weights to the primary inputs functionally dependent on the vector of contextual inputs $\overline{\mathbf{Z}}$. Consider a neuron model of the form

$$y = \Phi\left[w_0\left(\overline{\mathbf{Z}}\right) + \sum_{s=1}^{S} w_s\left(\overline{\mathbf{Z}}\right) x_s\right] \tag{1}$$

where x_s denotes the s-th primary input, y is the neuron's output, $\overline{\mathbf{Z}}$ denotes the vector of contextual variables, while Φ is the activation function. The neuron's weights to the primary inputs, and the offset, depend on the vector of contextual variables as follows:

$$w_s = w_s\left(\overline{\mathbf{Z}}\right) = \sum_{m=1}^{M} a_{s,m} v_m\left(\overline{\mathbf{Z}}\right) = \overline{\mathbf{A}}_s^T \overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right) \tag{2}$$

$s = 0, 1, \ldots, S$, where $\overline{\mathbf{V}}(\overline{\mathbf{Z}})$ is the vector of independent basis functions, common for all the net, spanning the weights' dependence on the context, and $\overline{\mathbf{A}}_s$ is the vector of coefficients for weight $w_s(\overline{\mathbf{Z}})$. The model is presented in fig. 1, where also the Kronecker product of primary input vector and the vector of basis functions of contextual inputs is shown.

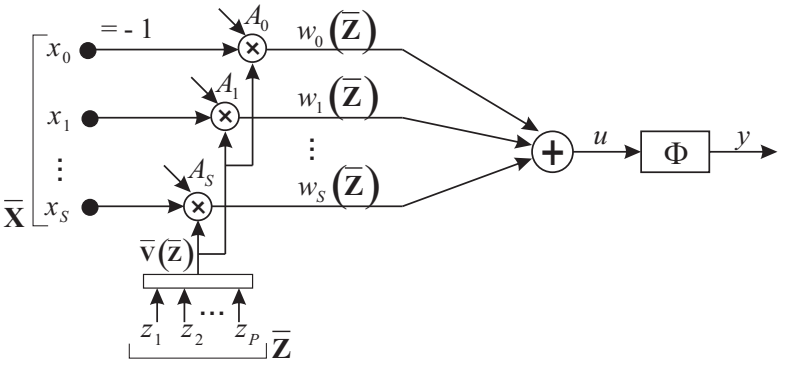

Fig. 1. Context-dependent neuron

2.2 Architectures of Context-Dependent Nets

A neural net may be used to solve a problem characterized both by primary and context-sensitive features. The former are useful for solving the task even when considered in isolation, regardless of their context. The latter require adapting the way of processing them to the contextual data. Therefore one may think of a hybrid net, in which some weights (for connections to context-sensitive inputs) are context-dependent and others (for connections to primary inputs) are traditional (constant after training, that is the same in all the contexts). If some additional knowledge of the problem being solved suggests using traditional weights, the net's designer may simply set some of the basis functions or coefficients for these weights to zero. It is also advisable to analyze the weights' change throughout the contexts when the learning process is completed in order to recognize the weights which should be constant. Different structures of hybrid nets are presented in [1].

2.3 Context-Dependent Feedforward Net

As in traditional nets, one can build more complicated structures made of single context-dependent neurons, particularly the common feedforward net. The parameters of context-dependent nets are stored in the vectors of coefficients for

each weight. It is advisable to construct the weight matrix for the whole layer of neurons in the following way:

$$\mathbf{A} = \left[\overline{\mathbf{A}}_1 \, \overline{\mathbf{A}}_2 \cdots \overline{\mathbf{A}}_K \right] = \begin{bmatrix} \overline{\mathbf{A}}_{1,0} & \overline{\mathbf{A}}_{2,0} & & \overline{\mathbf{A}}_{K,0} \\ \overline{\mathbf{A}}_{1,1} & \overline{\mathbf{A}}_{2,1} & & \overline{\mathbf{A}}_{K,1} \\ \vdots & \vdots & \cdots & \vdots \\ \overline{\mathbf{A}}_{1,S} & \overline{\mathbf{A}}_{2,S} & & \overline{\mathbf{A}}_{K,S} \end{bmatrix}_{(S+1)M \times K} \tag{3}$$

where $\overline{\mathbf{A}}_k$ is the vector of coefficients for the k-th neuron, built of vectors $\overline{\mathbf{A}}_{k,s}$ of coefficients for the weight $w_{k,s}\left(\overline{\mathbf{Z}}\right)$ (that is k-th neuron's weight to the s-th input). Such an organization of the net's parameters makes it possible to write the output of the net's layer $\overline{\mathbf{Y}}$ in a clear form as:

$$\overline{\mathbf{Y}}\left(\overline{\mathbf{X}}, \overline{\mathbf{Z}}\right) = \Phi\left\{ \overline{\mathbf{A}}^T \cdot \left[\overline{\mathbf{X}} \otimes \overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right) \right] \right\} \tag{4}$$

3 Properties of Context-Dependent Nets

The properties of traditional and context-dependent nets shall be compared in this section. The Vapnik-Chervonenkis dimension of both types of learning machines and the discriminating boundaries produced by them in their input space will be considered.

3.1 The Vapnik-Chervonenkis Dimension of CD-Nets

The Vapnik-Chervonenkis dimension is one of the quantities used in machine learning theory for estimating their generalization abilities and the effectiveness of learning. The idea of the growth function and VC-dimension is well presented in [10]. The parallel theory of the VC-dimension for context-dependent nets is developed in [1]. Here we shall present the main results and conclusions.

The VC-dimension of a traditional perceptron using S primary inputs is equal to the number of its adjustable parameters, that is $S + 1$.

In non-contextual approach the traditional perceptron is supplied with only primary features of the data (it is common for the donors of many datasets to exclude contextual information as irrelevant). Then a neuron with S inputs is able to dichotomize $S + 1$ points in its S-dimensional input space. If we also supply the neuron with information about the context of these points (encoded on P inputs and added to the S primary inputs already being used), then the neuron is able to dichotomize $S + P + 1$ points in the joint input space. In both cases the neuron's parameter space is of the same dimensionality as its input space.

The Vapnik-Chervonenkis dimension of a real weight context-dependent perceptron with $S \in N$ real primary inputs, using the basis function vector $\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right)$ consisting of M independent basis functions $v_m\left(\overline{\mathbf{Z}}\right)$, $m = 1, 2, \ldots, M$, is given by

$$\mathrm{VCdim}\left(H^{\mathrm{cont}}\right) = M\left(S + 1\right) \tag{5}$$

This result follows from the fact that for the above context-dependent neuron it is possible to find $M(S+1)$ (at most $S+1$ in the same context, that is with the same contextual vector $\overline{\mathbf{Z}}$) points in its input space, for which the neuron is able to perform all the dichotomies.

3.2 Separating Power of Traditional and CD-Nets

The separating abilities of both traditional and context-dependent neurons are strictly related to their parameter space. The dimensionality of the traditional neuron's parameter space is equal to the number of weights W. Analogically, for the context-dependent neuron it is equal to the number of coefficients $A = MW$.

The context-dependent neuron produces a discriminating hyperplane in its A-dimensional, that is $M(S+1)$-dimensional *parameter* space. This hyperplane is in fact a hypersurface in the $S+P+1$-dimensional *input* space. This leads to much more adjustable decision boundaries which context-dependent multilayer nets are able to produce in the joint input space. Therefore they are more powerful in classification tasks of contextual nature. Examples of discriminating hypersurfaces in a joint input space (including two primary and one contextual features) are presented in fig. 2. The hypersurfaces are lines for a fixed value of the contextual variable, as context-dependent neuron is a generalization of the traditional one and in the fixed context has the same properties. These discriminating lines may however much more flexibly change from one context to another.

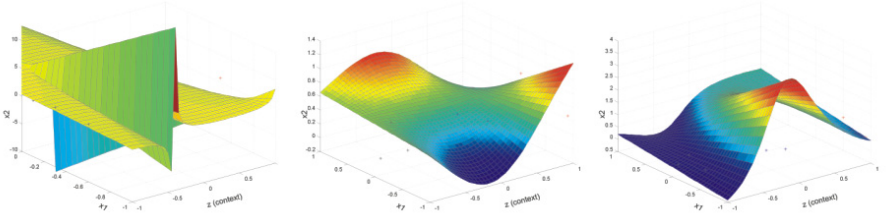

Fig. 2. Examples of discriminating hypersurfaces of context-dependent nets

The discriminating boundaries of traditional neurons are only hyperplanes in the joint input space. Nonlinear boundaries, more complicated in the contextual domain, must be modeled by a mixture of hyperplanes, that is by more neurons in the hidden layer of multilayer traditional net. Therefore both the decision boundaries, as well as their change through the contexts, have to be modeled by hidden layer's neurons of a traditional net. In context-dependent multilayer nets these two tasks are divided between the primary structure of the net (the weights) and the contextual structure (dependence of weights on the context).

Another advantage of context-dependent nets is the fact that contextual information modifies the weights of all the layers, modifying the behavior of the whole net according to the context. In traditional ones contextual data is supplied only to the first layer.

The Vapnik-Chervonenkis dimension of multilayer CD-nets grows similarly to traditional nets, that is with the number of net's parameters. However, the separating abilities of context-dependent nets are more evenly distributed between different contexts than for a traditional net of the same VC-dimension. The net's designer may also increase the VC-dimension by adding more basis functions. Not the growth of the VC-dimension should however be emphasized, but the effective use of the parameters space in order to obtain good generalization abilities of context-dependent nets in all the contexts.

As the interest in improving the learning machines performance by using contextual information grows, the models of context-dependent neural nets prove to be effective in solving tasks with contextual dependencies between the input data. An example of using a context-dependent neural net to modeling a highly nonlinear magnetorheological damper may be found in [11].

4 Training Context-Dependent Nets

Two algorithms for training multi-layer feedforward context-dependent nets are presented in this section: the backpropagation rule and contextual version of the Levenberg-Marquardt algorithm based on calculation of the error function's Hessian. A more thorough analysis of neural nets' training algorithms may be found in [1].

4.1 Context-Dependent Training with Backpropagation

Due to the appropriate construction of the net's coefficient matrix, the gradient based training algorithms are clear and their computational complexity is comparable to the one of traditional nets. For the net's output given by (4) the gradient (with respect to the coefficient matrix) of the following error function

$$E_{(\overline{\mathbf{X}},\overline{\mathbf{Z}},\overline{\mathbf{Y}})}\left[\Phi^{-1}\left(\overline{\mathbf{Y}}\right)-\mathbf{W}^{T}\left(\overline{\mathbf{Z}}\right)\cdot\overline{\mathbf{X}}\right]^{2}=E_{(\overline{\mathbf{X}},\overline{\mathbf{Z}},\overline{\mathbf{Y}})}\left\{\Psi\left(\overline{\mathbf{Y}}\right)-\mathbf{A}^{\mathbf{T}}\cdot\left[\overline{\mathbf{X}}\otimes\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right)\right]\right\}^{2} \tag{6}$$

where E is the expectation with respect to random variables specified below this operator, is given by

$$\nabla_{\mathbf{A}}E_{(\overline{\mathbf{X}},\overline{\mathbf{Z}},\overline{\mathbf{Y}})}=-\left\{\Psi\left(\overline{\mathbf{Y}}\right)-\mathbf{A}^{\mathbf{T}}\cdot\left[\overline{\mathbf{X}}\otimes\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right)\right]\right\}\cdot\left[\overline{\mathbf{X}}\otimes\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}\right)\right] \tag{7}$$

The backpropagation algorithm of training multi-layer context-dependent nets is similar to the one for traditional nets. It is derived and analyzed in detail in [1]. Here we shall give the final formula for the standard external error function's gradient (actual for the output layer's neurons and estimated for the hidden layers' neurons). The gradient is given as a vector for all neurons in the l-th

layer of the net (gradient of the error function E with respect to the l-th matrix coefficient matrix $A^{(l)}$, given by (3) for each layer):

$$\overline{\nabla}_{\mathbf{A}^{(l)}} E = \frac{\partial E}{\partial \mathbf{A}^{(l)}} = -\frac{1}{K_L}\left[\overline{\mathbf{D}}^{(l)} \circ \overline{\varphi'}^{(l)}\right] \otimes \left[\overline{\mathbf{X}}^{(l)} \otimes \overline{\mathbf{V}}\left(\overline{\mathbf{Z}}^{(l)}\right)\right] \tag{8}$$

where for the output layer $\overline{\mathbf{D}}^{(L)} = \overline{\mathbf{Y}}_d^{(L)} - \overline{\mathbf{Y}}^{(L)}$ and for the hidden layer $\overline{\mathbf{D}}^{(l)} = \mathbf{W}_{|\text{nobias}}^{(l+1)}\left(\overline{\mathbf{Z}}^{(l+1)}\right) \cdot \overline{\mathbf{D}}^{(l)} \circ \overline{\varphi'}^{(l+1)}$ L is the number of layers in the net, while K_l is the number of neurons in the l-th layer and $\overline{\mathbf{Y}}_d^{(L)}$ is the vector of desired outputs, $\overline{\varphi'}^{(l)}$is the vector of l-th layer neurons' activation functions' derivative.

4.2 Efficient Training with Hessian-Based Algorithms

The use of the Kronecker product in the above formulas is the key for developing effective Hessian-based training algorithms for context-dependent nets. One of them is the contextual version of Levenberg-Marquardt algorithm derived in [1]. Here we shall present the main results. The approximation of the inverse of the Hessian matrix (with respect to the k-th neuron's coefficient vector) of the error function given by (6) for a sequence of N training examples may be calculated as

$$\mathbf{H}_{\overline{\mathbf{A}}_{k,N}}^{-1} = \frac{1}{N}\left\{\sum_{n=1}^{N} e_{k,n}^2 \cdot \left[\overline{\mathbf{X}}_n \otimes \overline{\mathbf{V}}\left(\overline{\mathbf{Z}}_n\right)\right] \cdot \left[\overline{\mathbf{X}}_n \otimes \overline{\mathbf{V}}\left(\overline{\mathbf{Z}}_n\right)\right]^T\right\}^{-1} \tag{9}$$

where subscript n denotes a value for the n-th training example, $e_{k,n}$ is the k-the neuron's error. If all the examples from the sequence come from the same context group (have the same or similar values of the vector of basis functions, here denoted as $\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}_N\right)$), we may approximate the Hessian's inverse as

$$\mathbf{H}_{\overline{\mathbf{A}}_{k,N}}^{-1} = \frac{1}{N}\left\{\sum_{n=1}^{N} e_{k,n}^2 \cdot \left[\overline{\mathbf{X}}_n \cdot \overline{\mathbf{X}}_n^T\right]\right\}^{-1} \otimes \left[\overline{\mathbf{V}}\left(\overline{\mathbf{Z}}_N\right) \cdot \overline{\mathbf{V}}^T\left(\overline{\mathbf{Z}}_N\right)\right]^{-1} \tag{10}$$

The algorithm allows to reduce the computational complexity of training from $\mathcal{O}\left(W^3 M^3\right)$ to $\mathcal{O}\left(W^3 + M^3\right)$. As in most nets $M \ll W$, the computational effort of context-dependent net's learning is a little higher than for the traditional net of the same number of weights (thus the M times lower VC-dimension). The estimations give the possible training improvement of 10 to 1000 times (compared with traditional nets of the same VC-dimension, that is processing abilities) depending on the net's structure in traditional and contextual domain (the proportion of the number of weights to the number of basis functions). Let us notice that this computational gain is achievable only by applying the appropriate training algorithms and does not include the better convergence of net's parameters, expected from the better fitting of net's model to the model of the problem.

The presented algorithm is simple to apply as it is based just on the proper organization of training data into contextual groups containing examples of the

same (or comparable) value of contextual variables. Using samples from the same group in each training epoch (while mixing the examples from different groups between epochs) we may substitute the most operation demanding process of inverting the Hessian matrix with the inversion of two smaller matrices.

5 Conclusions

It has been shown that context-dependent neural nets, being the generalization of traditional nets, have better transformation abilities and are able to use their Vapnik-Chervonenkis dimension more efficiently than just with the growth of the net's size. The gradient-based training algorithms presented in the paper are of the comparable computational complexity than the ones for traditional nets. The original algorithms, using the properties of the Kronecker product for Hessian inverse calculation, improve the efficiency of training.

Acknowledgment. The work is supported by KBN grant in years 2002-2005

References

1. P. Ciskowski. *Learning of context-dependent neural nets.* PhD thesis, Wroclaw University of Technology, 2002.
2. P. Turney. The identification of context-sensitive features: A formal definition of context for concept learning. In *Proc. of 13th International Conference on Machine Learning (ICML96), Workshop on Learning in Context-Sensitive Domains*, Bari, Italy, 1996.
3. P. Turney. The management of context-sensitive features: A review of strategies. In *Proc. of 13th International Conference on Machine Learning (ICML96), Workshop on Learning in Context-Sensitive Domains*, Bari, Italy, 1996.
4. E. Rafajłowicz. Context dependent neural nets – problem statement and examples. In *Proc. of the Third Conference Neural Networks and Their Applications*, Zakopane, Poland, May 1999.
5. E. Rafajłowicz. Learning context dependent neural nets. In *Proc. of the Third Conference Neural Networks and Their Applications*, Zakopane, Poland, May 1999.
6. D. Yeung and G. Bekey. Using a context-sensitive learning for robot arm control. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1441–1447, Scottsdale, Arizona, May 14-19 1989.
7. S. Lee and G. A. Bekey. Application of neural networks to robotics. *Control and Dynamic Systems*, 39:1–67, 1991.
8. D.-T. Yeung and G.A. Bekey. On reducing learning time in context-dependent mappings. *IEEE Transactions on Neural Networks*, 4(1):31–42, 1993.
9. R. Watrous and G. Towell. A patient-adaptive neural network ECG patient monitoring algorithm. In *Computer in Cardiology*, Vienna, Austria, September 10-13 1995.
10. M. Anthony and P.L. Bartlett. *Neural Network Learning: Theoretical Foundations.* Cambridge University Press.
11. P. Ciskowski. Context-dependent neural nets in contextual modelling. In *Proc. International Conference on Signals and Electronic Systems, ICSES 2002*, Swieradow-Zdroj, Poland, 2002.