Forecasting Fetal Heartbeats with Neural Networks¹

Claudia Ulbricht, Georg Dorffner Austrian Research Institute for Artificial Intelligence, Schottengasse 3 and Institute of Medical Cybernetics and Artificial Intelligence, Freyung 6 A-1010 Vienna, Austria claudia@ai.univie.ac.at, georg@ai.univie.ac.at

> Andreas Lee Department of Prenatal Diagnosis and Therapy University Hospital Vienna Währinger Gürtel 18–20, A–1090 Vienna, Austria A–1090 Vienna, Austria Andreas.Lee@akh-wien.ac.at

Abstract

The given task is to forecast the intervals between the heartbeats recorded from a fetus. The six tested neural network models combine input windows, hidden layer feedback, and self-recurrent unit feedback in different ways. The two networks combining an input window and hidden layer feedback performed best. One of them has additional self-recurrent feedback loops around the units in the state layer, which enable the system to deal with time-warped patterns. It turns out to be reasonable to combine several techniques for processing the temporal aspects inherent to the input sequence.

1 The Task

Using the cardiotocogram (the CTG) is common for routine fetal monitoring. The CTG consists of fetal heartbeat and uterine contraction signals. At the site under investigation, such signals have been recorded and stored for further analysis. Usually, the heart rate is pre-processed before it is analyzed. In this study, though, each single heartbeat interval is recorded for obtaining greater precision. The overall aim is the development of an intelligent alarm system which can be employed as a tool for decision support. The first step when processing the

¹This is an extended version of the paper: Ulbricht C., Dorffner G., Lee A.: Forecasting Fetal Heartbeats with Neural Networks, in Bulsari A.B., et al.(reds.), Solving Engineering Problems with Neural Networks, Systeemitekniikan seura ry, Turku, pp.403-406, 1996.

given data sets is to detect the artefacts, so that they can be removed. In order to improve the detection of artefacts, the next value in the time series can be forecast and compared with the actual value. Values which deviate considerably from the forecast are more likely to be disturbed by measuring errors than those which are close to the forecast ones. As proposed in [Miksch *et al.*, 1995], such a forecasting system could also be used for "repairing" the input signals. Instead of replacing missing values by average or preceding ones, they could be replaced by the forecast values which are more likely to resemble the true values.

2 Tested Neural Network Models

Most neural network research has focused on processing single patterns, but sequence processing requires a method for saving information for subsequent time steps. Overviews of such neural networks, which can be used for handling temporal aspects, are given, for instance, in [Ulbricht *et al.*, 1992], [Mozer, 1993], [Rohwer, 1994], and [Chappelier and Grumbach, 1994]. All the networks tested on the task of forecasting heartbeat intervals had at least a single input unit for the sequence element, three hidden units, and one output unit for the forecast. The six tested variants of network models are depicted in Figs. 1 through 3. The focus lies on the layers and the links between them. The numbers of the layers refer to the order in which they are updated. A dashed arrow denotes a link for copying unit activations, whereas a full arrow denotes a set of links connecting each unit of one layer with all the units of the other layer. The following models were tested:

- 1. A network with an input window (Fig. 1): In this non-recurrent network, the window is obtained by delaying the single sequence element at the input several times in a row. The resulting window is of size 5, i.e. it contains the five most recent sequence elements $x(t-1), x(t-2), \ldots, x(t-5)$.
- 2. A network with hidden layer feedback (Fig. 1): In this simple recurrent network, the hidden layer is delayed and fed back like in the network described in [Elman, 1990].
- 3. A network with an input window of size 5 and hidden layer feedback (Fig. 2): It combines delays with and without feedback.
- 4. A network with a self-recurrent feedback loop around the input (Fig. 2): In this network, the temporal aspects are handled by delaying and feeding back the activation of a unit to itself. The feedback loop has a weight μ , and the input is weighted by $1-\mu$.
- 5. A network with an input window of size 5 and self-recurrent feedback loops around all the units in the input window (Fig. 3): It uses both a window and unit feedback.

6. A network with an input window of size 5, hidden layer feedback, and self-recurrent feedback loops around the units in the memory layer (Fig. 3): In the taxonomy presented in [Mozer, 1993], such a memory with self-recurrent feedback loops is referred to as "exponential trace memory," because it contains an exponentially weighted average. This memory can also be regarded as the state of the network. Due to the feedback loops, the state changes more slowly. The speed of change depends on the weights. Such sluggish states can also be obtained by using any other nearly auto-associative next-state function. The important point is that "state vectors at nearby points in time must be similar," as it is stated in [Jordan, 1986]. The resulting models are better suited to dealing with patterns in sequences which are warped in time, because they have the intrinsic capability of generalizing over the temporal dimension.



Figure 1: Networks 1 and 2



Figure 2: Networks 3 and 4

In all the networks, the weights μ of the self-recurrent feedback loops were equal to 0.9. The forecast of the window network can be written as:

$$\hat{x}(t) = F_1\left(x(t-1), x(t-2), x(t-3), x(t-4), x(t-5)\right).$$
(1)

Thereby F_1 denotes the mapping of the whole neural network. The forecast of the network with hidden layer feedback is

$$\hat{x}(t) = F_2(x(t-1), \mathbf{h}_1(t-1)),$$
(2)



Figure 3: Networks 5 and 6

where $\mathbf{h}_1(t-1)$ is a vector of length 3 (bold letters are used for vectors and mappings to vectors). Since the hidden layer is part of the feedback loop, it contains information of past inputs:

$$\mathbf{h}_{1}(t-1) = \mathbf{f}_{1}(x(t-2), \mathbf{h}_{1}(t-2)), \qquad (3)$$

where \mathbf{f}_1 represents the mapping from the input to the hidden layer. The forecast of the third network is:

$$\hat{x}(t) = F_3(x(t-1), x(t-2), x(t-3), x(t-4), x(t-5), \mathbf{h}_2(t-1)), \qquad (4)$$

where $\mathbf{h}_2(t-1)$ is:

$$\mathbf{h}_{2}(t-1) = \mathbf{f}_{2}\left(x(t-2), x(t-3), x(t-4), x(t-5), x(t-6), \mathbf{h}_{2}(t-2)\right).$$
(5)

The fourth network has unit feedback in the input layer:

$$\hat{x}(t) = F_4\left((1-\mu) \ x(t-1) + \mu \ i_{1_1}(t-1)\right),\tag{6}$$

where $i_{1_1}(t)$ stands for the single element of the contents of the input layer $i_1(t)$:

$$i_{1_1}(t) = (1-\mu) \ x(t-1) + \mu \ i_{1_1}(t-1).$$
(7)

For the network with an input window and self-recurrent feedback loops, the output is:

$$\hat{x}(t) = F_5((1-\mu) x(t-1) + \mu i_{2_1}(t-1),
(1-\mu) x(t-2) + \mu i_{2_2}(t-1),
(1-\mu) x(t-3) + \mu i_{2_3}(t-1),
(1-\mu) x(t-4) + \mu i_{2_4}(t-1),
(1-\mu) x(t-5) + \mu i_{2_5}(t-1)).$$
(8)

In this equation, $i_{2_1}(t-1), \ldots, i_{2_5}(t-1)$ are the components of the vector $\mathbf{i}_2(t-1)$ representing the input layer. Finally, the forecast obtained with the sixth network

can be described as:

$$\hat{x}(t) = F_6(x(t-1), x(t-2), x(t-3), x(t-4), x(t-5),
(1-\mu) h_{3_1}(t-1) + \mu s_1(t-1),
(1-\mu) h_{3_2}(t-1) + \mu s_2(t-1),
(1-\mu) h_{3_3}(t-1) + \mu s_3(t-1)),$$
(9)

where $h_{3_1}(t-1), \ldots, h_{3_3}(t-1)$ are the components of the hidden layer $\mathbf{h}_3(t-1)$, and where $s_1(t-1), \ldots, s_3(t-1)$ are the components of the state layer $\mathbf{s}(t-1)$.

3 Comparative Analysis

A sequence consisting of 1200 elements was used as the training set. The validation set, which contained 600 elements, was used to determine when to stop training. Another 600 sequence elements were used for testing. A segment of the sequence of heartbeats is depicted in Fig. 4. The heartbeats were measured in milliseconds. The interval ranging from 0 to 1200 milliseconds was transformed for the network to the interval ranging from zero to one. The mean square error (MSE) is taken as a measure for evaluating the performance:

MSE =
$$\frac{1}{N} \sum_{n=1}^{N} (x_d(n) - x_t(n))^2$$
, (10)

where $x_d(n)$ is the *n*-th network output, and $x_t(n)$ the *n*-th target output out of N instances. An auto-regressive model, an AR[1] model as described in [Box and Jenkins, 1970],

$$x(t) = \alpha x(t-1) + \varepsilon(t), \tag{11}$$

is set up for comparison with the networks. With α equal to one, a random walk process is modeled. When using this model for forecasting, the estimate for x(t)is equal to x(t-1). If it turns out that this is the best estimate, it can only be said that subsequent intervals are likely to be similar. However, if better forecasting models can be found, more can be said about the sequence of beats. For such an AR[1] model with an α equal to one, the MSE on the test set is 0.0047. Each type of network was tested three times. The results are visualized in Fig. 4. For each network, the MSE averaged over three tests with random weight initialization is shown.

Each type of network was tested three times. The MSE on the test set is given in Table 1 for all the experiments. Additionally, the mean of the three experiments is provided. The results are visualized in Fig. 4.

It turns out that it is possible to obtain better forecasts with appropriately designed neural networks than with a simple AR[1] model. The networks combining an input window and hidden layer feedback (Networks 3 and 6) perform best. According to the t-Test, the results of these two networks are significantly

Fetal Heartbeat Intervals



Figure 4: Sequence of heartbeats



Figure 5: Overview of the results from forecasting heartbeat intervals

	Configuration			MSE			
Net Nr.	Window Size	Hidden Feedback	Self-recurrent Feedback	Test 1	Test 2	Test 3	Mean Error
$\begin{array}{c}1\\2\\3\\4\\5\\6\end{array}$	$5 \\ 1 \\ 5 \\ 1 \\ 5 \\ 5 \\ 5$		Input Window Memory	$\begin{array}{c} 0.0066\\ 0.0078\\ 0.0039\\ 0.0083\\ 0.0077\\ 0.0038\\ \end{array}$	$\begin{array}{c} 0.0053\\ 0.0060\\ 0.0038\\ 0.0071\\ 0.0077\\ 0.0038\end{array}$	$\begin{array}{c} 0.0064\\ 0.0039\\ 0.0041\\ 0.0066\\ 0.0073\\ 0.0038\\ \end{array}$	$\begin{array}{c} 0.0061\\ 0.0059\\ 0.0039\\ 0.0073\\ 0.0076\\ 0.0038\\ \end{array}$

Table 1: Results from forecasting heartbeat intervals

better at the 95%-level. Moreover, the following points result from an analysis of the mean errors:

- Network 3, having both an input window and hidden layer feedback, performs very well in all the experiments. This demonstrates how an appropriate combination of non-recurrent and recurrent mechanisms (an input window and hidden layer feedback) can lead to much better results than using only one of the two mechanisms, as it is the case in Networks 1 and 2.
- Self-recurrent feedback loops in the input layer, as they are used in Networks 4 and 5, are not well suited to this application. They do not suffice to provide enough information about the sequence. This is a typical result, as self-recurrent input feedback alone is not sufficient for capturing the temporal aspects which are relevant to most types of applications.
- The network which combines several sequence-handling methods (Network 6) performs even slightly better than the same type of network without self-recurrent feedback loops (Network 3). This shows that unit feedback in the state layer, which leads to a slowly changing state, can improve the performance of the network.

4 Conclusion

The given task was to forecast the intervals between fetal heartbeats. The performances of six different neural network models and a simple auto-regressive model were tested empirically. The outcome can be regarded as an example demonstrating that good results can be obtained when various methods are combined in a single neural network. The best performance was obtained with a network which used layer delay, layer feedback, and unit feedback. It can be seen that the outcome is heavily dependent on how the state is formed. Additional self-recurrent feedback loops around the state layer can even slightly improve the network performance. They make the state change more slowly, which is better for dealing with time-warped sequences. Even though the results are different for each application, it can be concluded that combining several techniques for processing the temporal aspects inherent to the input sequence seems to be reasonable. Especially the combination of an input window and layer feedback turns out to lead to good results. They can function better than an input window or layer feedback alone.

Acknowledgements

This research is supported by the "Medizinisch–Wissenschaftlicher Fonds des Bürgermeisters der Bundeshauptstadt Wien."

References

- [Box and Jenkins, 1970] G.E. Box and G.M. Jenkins. *Time Series Analysis*. Holden-Day, San Francisco, 1970.
- [Chappelier and Grumbach, 1994] J.-C. Chappelier and A. Grumbach. Time in Neural Networks. SIGART Bulletin, Vol. 5, No. 3, pages 3-11, 1994.
- [Elman, 1990] J.L. Elman. Finding Structure in Time. Cognitive Science, 14:179– 211, 1990.
- [Jordan, 1986] M.I. Jordan. Attractor Dynamics and Parallelism in a Connectionist Sequential Machine. In Proceedings of the Eighth Annual Conference of the Cognitive Science Society, pages 531-546. Erlbaum, Hillsdale, NJ, 1986.
- [Miksch et al., 1995] S. Miksch, W. Horn, C. Popow, and F. Paky. Automated Data Validation and Repair Based on Temporal Ontologies. Technical Report TR-95-04, Österreichisches Forschungsinstitut für Artificial Intelligence, Wien, 1995.

- [Mozer, 1993] M.C. Mozer. Neural Net Architectures for Temporal Sequence Processing. In A. Weigend and N. Gershenfeld, editors, *Predicting the Future* and Understanding the Past. Addison-Wesley Publishing, Redwood City, CA, 1993.
- [Rohwer, 1994] R. Rohwer. The Time Dimension of Neural Network Models. SIGART Bulletin, Vol. 5, No. 3, pages 36-44, 1994.
- [Ulbricht et al., 1992] C. Ulbricht, G. Dorffner, S. Canu, D. Guillemyn, G. Marijuán, J. Olarte, C. Rodriguez, and I. Martin. Mechanisms for Handling Sequences with Neural Networks. In C.H. Dagli et al., editors, *Intelligent Engineering Systems through Artificial Neural Networks*, ANNIE'92, volume 2, pages 273-278. ASME Press, New York, 1992.