WarpNet: Self-Organizing Time Warping

Kari Torkkola

Motorola, Phoenix Corporate Research Laboratories,
2100 East Elliot Road, MD EL508, Tempe, AZ 85284, USA
tel: (602)413-4129, fax: (602)413-7281, email: A540AA@email.mot.com

Abstract

We describe "WarpNet", a time-warping algorithm for speech recognition based on neural nets. WarpNet is a self-organizing matching mechanism intended to replace dynamic programming (Dynamic Time Warping or Viterbi-algorithms). The concept is based on elastic models that self-organize in the spirit of Self Organizing Maps and Optimizing Maps [9] so as to minimize the the sum of distances over two utterances, or any time series. We show that both the recognition accuracy, and the computing time of WarpNet are comparable to the optimal matching method, dynamic programming.

1 Introduction

Neural nets have traditionally not been among the most popular tools in handling the time dimension in signal processing. In any speech recognition system that uses neural nets as components the time alignment has always been done by some variant of dynamic programming, either by Dynamic Time Warping (DTW), or in systems based on Hidden Markov Models (HMM), by the Viterbi-algorithm. Invariably, neural nets appear in some other role such as vector quantizers or discriminative probability generators.

Given two discrete-time signals, the template and the input, the time alignment problem can be stated as follows: Align the signals so as to minimize the sum of samplewise distances between them. Samples can be n-dimensional vectors such as short-time spectra or cepstra computed from speech signals.

It is actually not surprising that there has not been too much research on applying neural nets to this optimization problem (which is not even np-hard), because there exists an optimal solution based on dynamic programming, which is called Dynamic Time Warping. DTW works as follows [12]. Denoting the length of the template as L_t and length of the input as L_i, a $L_t \times L_i$ matrix of all mutual samplewise distances is first computed. Then a path of smallest accumulated distance through the matrix is searched by dynamic programming. This is essentially a sequential process. Computational complexity of the method is $O(L_t \times L_i)$.

Some attempts have been made to map the DTW problem or the Viterbi algorithm onto Hopfield nets [14, 4]. These approaches generally require a number of "neurons" of $O(L_t \times L_i)$, and generally, an order of magnitude larger number of connections. They will thus not be very feasible in any larger task.

Other approaches include [7], which introduces "time-warping neurons" in a multilayer structure. But these neurons are larger entities that actually utilize DTW locally to perform the warp. In [8] a prewarp is performed before recognition. A neural net is trained to make a local judgement about the speed of speech production. Neither method is thus intended as an alternative mechanism to DTW.

Some network architectures, such as time-delay neural networks can provide slight tolerance to time shifts, but these methods have also been used in conjunction with the dynamic programming methods for time alignment [5].

Since DTW is essentially a sequential process, we wanted to explore possibilities for parallelism using neural networks of self-organizing type. This led us to examine elastic models, that have been widespread in character and object recognition. The rest of this paper is structured as follows. We will proceed by describing some of the approaches in elastic model literature. Then we will explain the principles behind our approach, "WarpNet". Finally, we will describe some speech recognition experiments with WarpNet, and we will conclude by pointing out connections to the image processing literature, as well as some of the shortcomings and advantages of WarpNet as compared to DTW.

2 Elastic models in image and character recognition

The method we will present bears a strong resemblance to elastic models for object recognition [6, 1, 13, 2, 10] and for character recognition [3, 15, 9, 11]. These kind of models with iterative matching and annealing mechanisms have been of great interest, because there is no matching algorithm for 2-dimensional data corresponding to DTW. This is due to the fact that the problem is now np-hard (the problem of subgraph isomorphism).

Malsburg's Dynamic Link Architecture is one example of a neural net architecture for elastic graph and image matching. This has been described in, e.g., [6, 2], or in [13] with some extensions.

Williams et al. describe probabilistic generative models for character recognition [15, 11]. Their models have a large variance in the beginning corresponding to coarse matching, which decreases as the matching proceeds to finer details. Their actual matching mechanism is the Expectation-Maximization algorithm, trying to maximize the likelihood of data generated by the model, at the same time minimizing model deformations.

Asogawa describes an elastic input field to a character recognizer [3]. Each sample is drawn towards a direction (in 2-d space) that minimizes its distance to a sample of template. In addition, the samples in the sampling grid are held together by forces that represent the rigidity/elasticity constraints and order-preserving constraints. Asogawa proceeds to show that the sum of the forces is a Lyapunov function and thus the iterations converge.

However, the starting point of our work has been the so called Optimizing Maps [9]. These are character models as Self-Organizing Maps, in which the topology of each SOM model follows the topology of the actual character. Each model is matched to the input by a process that resembles normal SOM training. Dark pixels of the input image are the training data that attracts the model to overlay the structure of the input data. The model with the smallest distortion according to a certain “quality-of-fit” measure is chosen as the recognition result. While this approach works well in character recognition providing rotation, translation, scale, and local nonlinear deformation invariance, all these properties are not necessary in 1-d signal matching. Only deformation, scale, and to a minor extent, translation invariance are useful. Thus, to develop a computationally efficient algorithm, we decided to abandon the true SOM training in the matching process and to replace it with an appropriate shortcut as described in the following section.

3 Self-organizing time-warping

We will now describe the self-organizing matching mechanism of WarpNet. In our configuration, the template will be held fixed but the input will be translated and warped to match the template. For example, in isolated word speech recognition there will be several templates corresponding to different words of the vocabulary. The unknown input will be warped to match each of the templates in the best possible way, and the recognition tesult will be the template with the smallest sum of distances after the warp.

It is first determined which direction (left or right) to move each sample to make its distance smaller to the model. Each sample negotiates with its neighboring samples to determine a good direction for the whole neighborhood. Now SOM-like ideas enter into the picture. At first, neighborhood size includes all input samples. Progressively, neighborhood size decreases. At the end, each sample makes individual decisions. The process is illustrated in Fig. 1, which depicts two sequences, the template (thicker) and the input (thinner).

For each sample of the input, its distance to a template sample left to its current position and to the right of its current position is computed. Next, it is determined which would be a better direction to move the position of the input sample to make the distance smaller. This results in an array of directions with magnitudes, since also the amount how much smaller the distance would get will be calculated.

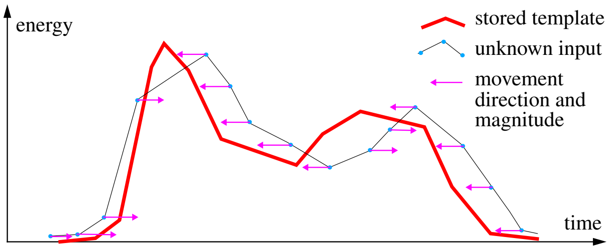

Figure 1: The principle of WarpNet.

In practice, the neighborhood effect will be achieved by filtering the direction array. The smoothing filter will be very wide in the beginning of the process (corresponding to a wide neighborhood in the SOM), and it will get gradually narrower and sharper. At the end of the process there will be no smoothing at all corresponding to zero neighborhood in the SOM algorithm. Thus, in effect the filtering of the direction array controls the elasticity of the input sequence during the matching process. A wide smoothing filter is essential in the beginning to ensure convergence to a global minimum.

The current time positions of the input samples are updated on the basis of the smoothed direction array. This is done simply by multiplying the direction array by a plasticity factor (which also gets smaller during the process) and by summing it to the array of current positions. Restrictions about the maximum amount of time scale compression and stretching are imposed at this point. Also endpoint alignment restrictions can be taken into account.

If the signals are badly misaligned to begin with, it is advantageous to perform the alignment in multiple scales of resolution. With speech feature vector sequences this can be achieved, for example, by low-pass filtering the sequences.

The whole matching process as applied to isolated words progresses now as follows:

1. Initialize. The easiest thing is just to warp the input linearly to have the same length as the template. In practice it would be best to isolate the speech portions of both signals, and only warp those.

2. Run a few iterations using a wide fixed smoothing filter and a fixed large plasticity. The purpose of this step is to take care of translation, to move the input as a whole over the template. This step is done at a coarser resolution with filtered template and input signals.

3. Switch back to non-filtered template and input signals. Run a few iterations while the smoothing filter gradually narrows down to zero, and the plasticity decreases. This is the actual organizing phase where the corresponding parts of the signals will be discovered.

4. Some fine tuning iterations with a small fixed plasticity and no smoothing of the movement directions.

Note that as opposed to DTW, the positions of the input samples will be real values, not discrete. To evaluate the distance of an input sample to the corresponding template sample, linear interpolation is used to find a value for the template at this given real-valued time instant.

Instead of interpolating a vector between two discrete time samples of the template and evaluating the distance to this interpolated vector, we can compute distances to two adjacent discrete time samples of the template, and interpolate between the distances. In general, this results in larger distances requiring smaller plasticity. This modification speeds up the process somewhat. Another speedup is achieved by tabulating already computed distances since at the end of the algorithm when the input moves/warps only a little at a time, these same distances will be used over and over.

4 Experiments and examples

To evaluate the accuracy of WarpNet in speech recognition, we compared it to DTW using the isolated digit part of the TIDIGITS database (male & female). DTW is the appropriate benchmark against WarpNet, since it provides the optimal match between two utterances. This is exactly what we also propose WarpNet for, simply as a matching mechanism between two utterances. DTW provides thus the baseline of best possible matching performance.

The training part of the database was used as templates, against which the 2486 test digits were compared, both using WarpNet and DTW. We did not use any k-nn (k nearest neighbors) schemes. The recognition result was simply the class of the closest template. Note that there is no training involved since we are just comparing two different matching mechanisms. In a real speech recognition system one would certainly like to reduce the number of templates by using any clustering or training method for the DTW.

As the feature vector representation of the utterances, we computed 10-component mel-scale cepstra each 10 ms from overlapping windows of 25.6 ms. Plain Euclidian distance was used by both methods as the distance measure between cepstral vectors. The maximum allowed time scale warping was limited between 2 and 0.5. The smoothing filter was Gaussian-shaped.

The results are presented in the table below.

	WarpNet	DTW
accuracy	99.12%	99.24%
number of errors	22	19

We can see that WarpNet makes slightly more errors (3/2486) than the optimal method, a difference which is insignificant. Both methods, written in highly optimized c-code took the same CPU time on a HP unix-workstation.

To furthermore illustrate the algorithm, we present in Fig. 2 a series of snapshots of WarpNet aligning two utterances of digit “three”. The feature representation of the utterances is 20-component short-time spectra for visualization. See figure caption for further explanation.

Fig. 3 illustrates the differences between WarpNet and DTW. The same aligment problem of Fig. 2 is solved using DTW. The large rectangle on the right depicts distances between each sample of the template (vertical axis) and the input (horizontal axis). The darker the color, the smaller the distance. The task of DTW is to find a path through this rectangle so that the sum of distances along the path is smallest possible. This path is plotted on top of the square. The alignment before and after applying DTW is plotted on the left. Two differences to WarpNet are apparent. DTW has the possibility of mapping two consecutive samples of the input to the same template sample, which WarpNet cannot do due to maximum compression restriction of a factor of 2. These mappings are seen as the negative peaks in the warping curve. The second difference is that WarpNet can place the samples of input template freely in time whereas DTW is restricted to the original discrete locations. This actually might give WarpNet an advantage in some cases. Here it can be seen as the smaller total distance along the warping path.

5 Discussion

The method we have presented bears a strong resemblance to elastic models for object recognition and character recognition. Asogawa's elastic input field [3] shares some features similar to WarpNet. The “force” that attracts similar samples in the template and the input is based on the intensity gradient, and resembles our array of movement directions and magnitudes. Asogawa's convergence proof approach could also easily be adapted to prove that WarpNet converges.

As opposed to Optimizing Maps [9], we abandoned the true SOM training in the matching process and replaced it with an appropriate shortcut where each input sample only evaluates distances in its immediate vicinity instead of searching the best match over the whole network. The shrinking neighborhood that provides the global ordering is, however, retained in the smoothing process of the WarpNet.

If DTW and WarpNet have about the equal performance why would one want to use WarpNet? At the moment the only advantage seems to be that WarpNet would be slightly easier to parallelize on special hardware. Whereas the search part of DTW is strictly a sequential process, all operations in a WarpNet iteration could be performed in parallel. Computational complexity of WarpNet is only $O(L_i)$ as opposed to $O(L_t \times L_i)$, but the constant in front of it is large at the moment.

Some limitations of WarpNet are that it relies on the smoothness/continuity of the data such as speech. It would not find the optimal alignment with random-like sequences, whereas DTW would. Also the recognition of connected or continuous speech is not addressed yet but we can foresee these problems being addressed in a similar fashion as in two-level, or level-building DTW.

References

[1] A.J. Abrantes and J.S. Marques. Unified approach to snakes, elastic nets and Kohonen maps. In *Proc. ICASSP*, pages 3427–30, Detroit, MI, May 9-12 1995. IEEE.

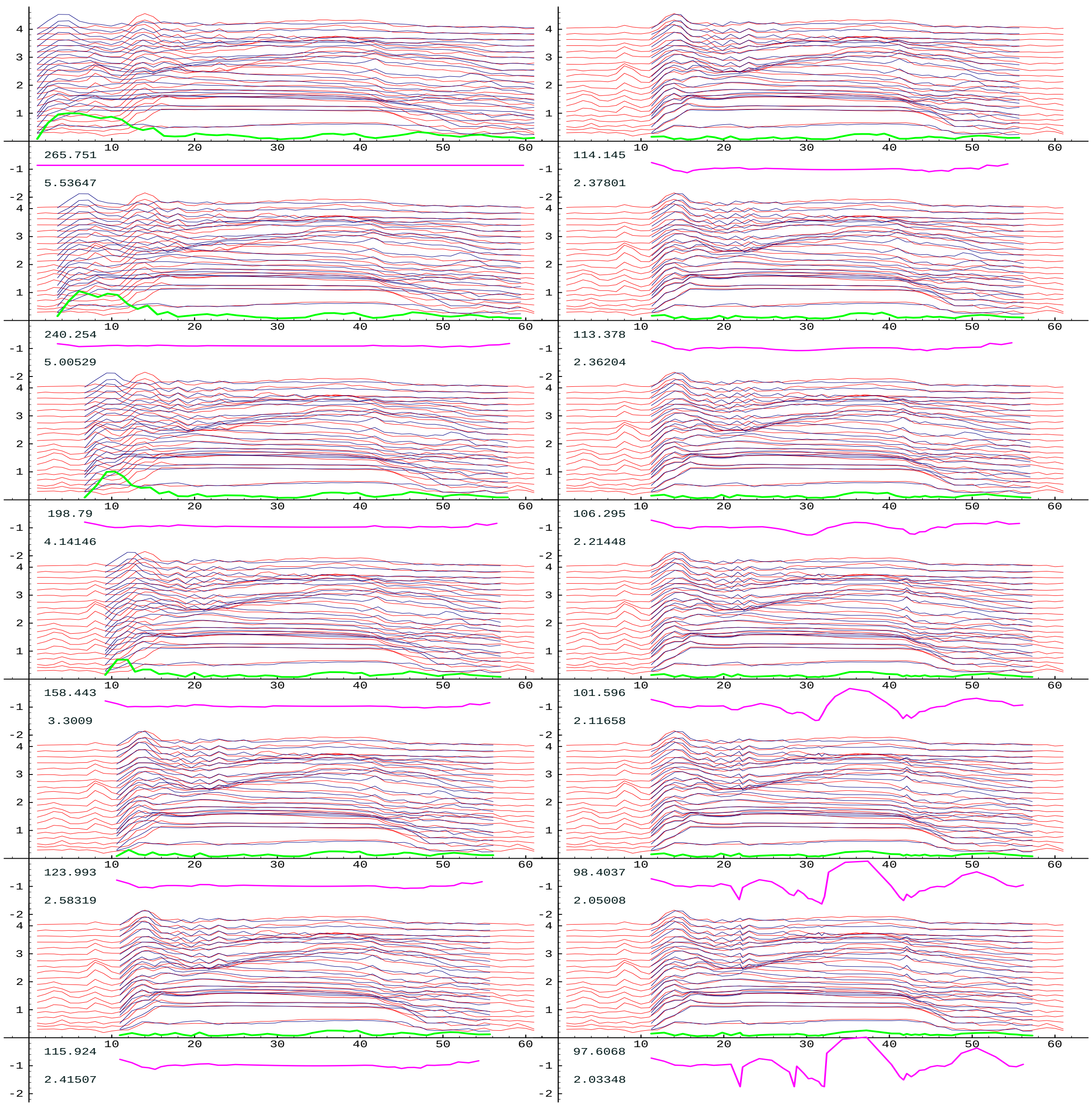

Figure 2: WarpNet matching two instances of digit "three" spoken by a female person. Read from top to bottom, from left to right. 20-component short time spectra of the utterances are plotted on top of each other. Each line represents energy in a frequency band, which are listed at the positive vertical axis in kHz. The single curve below the spectra represents warping of the input to match the fixed template. Values above -1 denote nonlinear stretching, values smaller than -1 denote nonlinear compression. The slightly thicker curve just above the horizontal axis displays the distances between feature vectors of the utterances as aligned. Upper number is the sum of these distances over the utterances (the criterion WarpNet is trying to minimize), and the lower number is the average distance between feature vectors.

[2] T. Aonishi and K. Kurata. Deformation theory of dynamic link architecture. *Neural Computation*, 1996. (submitted), also ftp://ftp.bpe.es.osaka-u.ac.jp/pub/FukushimaLab/Papers/aonishi/deform_dy.ps.gz.

[3] Minoru Asogawa. Adaptive elastic input field for recognition improvement. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in neural information processing systems 7*, pages 512–519. MIT Press, Cambridge, MA, 1995.

[4] Sreeram V. Balakrishnan-Aiyer. *Solving combinatorial optimization problems using neural networks*

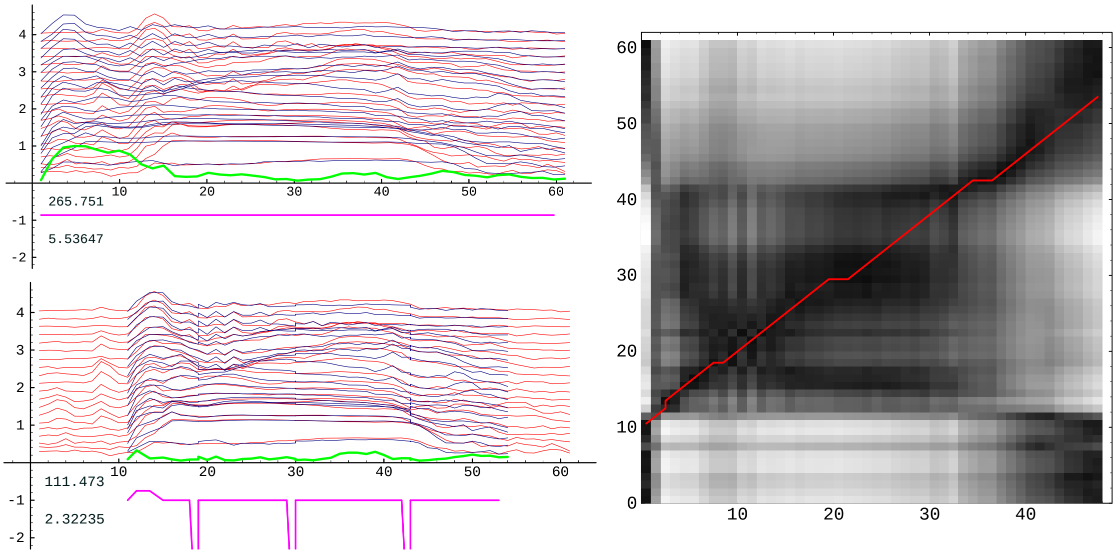

Figure 3: DTW aligning two instances of digit "three". See text.

with applications in speech recognition. PhD dissertation, University of Cambridge, October 1 1991. CUED/F-INFENG/TR.89.

[5] Patrick Haffner, Michael Franzini, and Alex Waibel. Integrating time alignment and neural networks for high performance continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP91)*, volume 1, pages 105–108, Toronto, Canada, May 14-17 1991.

[6] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. vd Malsburg, R.P. Würtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.

[7] Esther Levin, Roberto Pieraccini, and Enrico Bocchieri. Time-warping network: A hybrid framework for speech recognition. In John E. Moody, Stephen J. Hanson, and Richard P. Lippmann, editors, *Advances in neural information processing systems 4*, pages 151–158. Morgan Kaufmann, San Mateo, CA, 1992.

[8] Earl Levine. A time warping neural network. In *Proc. ICASSP*, pages 3339–3341, Detroit, MI, May 9-12 1995. IEEE.

[9] William M. Peterson, Jim McGuire, Kevin Reinhart, and Sidney C. Garrison III. Method and apparatus for recognition using a neural network, 1996. U.S. patent pending.

[10] J. Qian, T. Mitsa, and E.A. Hoffman. A physically based model for the registration of a 2D image sequence. In *Proc. ICASSP*, pages 2197–2200, Atlanta, GA, May 7-10 1996.

[11] Michael D. Revow, Christopher K. I. Williams, and Geoffrey E. Hinton. Using generative models for handwritten digit recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):592–606, 1996.

[12] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[13] Soheil Shams. Multiple elastic modules for visual pattern recognition. *Neural Networks*, 8(9):1439–1456, 1995.

[14] F.A. Unal and N. Tepedenlenligolu. Dynamic time warping using an artificial neural network. In *Proc. IJCNN'92, Int. Joint Conference on Neural Networks*, volume IV, pages 715–721, Baltimore, MD, June 7-11 1992. IEEE Service Center.

[15] Christopher K. I. Williams, Michael D. Revow, and Geoffrey E. Hinton. Using a neural net to instantiate a deformable model. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in neural information processing systems 7*, pages 965–972. MIT Press, Cambridge, MA, 1995.