BREATH DETECTION USING A FUZZY NEURAL NETWORK AND SENSOR FUSION

K. P. Cohen, Y. H. Hu, W. J. Tompkins and J. G. Webster
University of Wisconsin at Madison
Department of Electrical and Computer Engineering
1415 Johnson Drive
Madison, Wisconsin 53706
USA

ABSTRACT

We have developed and trained a Fuzzy Neural Network (FNN) to detect individual breaths using information from multiple independent noninvasive ventilation sensors. We derive input features from simultaneous recordings from impedance and inductance plethysmographs, and a pneumotachometer while healthy adults performed several different combinations of ventilation and motion. We first tested our FNN using membership functions, rules and consequent sets derived using a heuristic approach. Using all features, on 4 subjects we found that the average rate of combined false-positive and false-negative detections was 5.1%. When we trained our FNN using a gradient descent algorithm, the average rate of combined false-positive and false-negative detections was reduced to 2.6%.

1. INTRODUCTION

Accurate noninvasive breath detection is desirable in a wide variety of clinical and research applications. Several noninvasive sensing technologies have been developed. However artifacts resulting from subject motion, airway obstruction and electrocardiographic signals can make it difficult to detect breaths. False detection can have severe consequences. For example, in infant apnea monitors, if true breaths are not detected a false alarm will sound. This is extremely stressful to caretakers. Alternatively, false breath detections during periods without breathing can inhibit an alarm during a life threatening event, which can be fatal.

1.1. Noninvasive Ventilation Measurements

Impedance and inductance-based instrumentation are commonly used to obtain a noninvasive measure of ventilation. Impedance plethysmographs sense changes in transthoracic impedance between a pair of ribcage electrodes [1]. Inductance plethysmographs measure changes in the self-inductance of wires encircling the abdomen and ribcage, which approximate the changes in abdomen and ribcage cross-sectional area, respectively [2]. Pneumotachometers measure airflow, and can be used as a "gold standard" to determine sensor and algorithm performance (fig. 1).

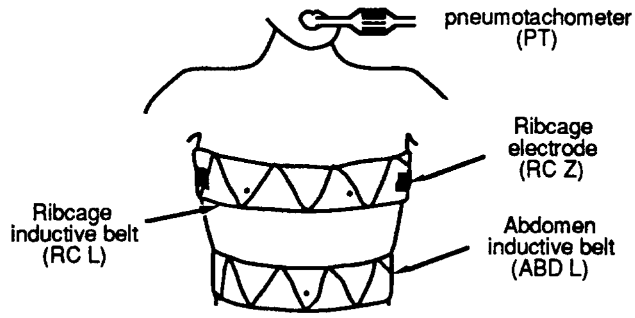

Fig. 1. Ventilation measurements.

During normal breathing, signals from impedance and inductance-based sensing technologies are qualitatively similar, and simple breath-detection algorithms can accurately detect breathing. However, movements unrelated to breathing or episodes of airway obstruction can appear as breathing (fig. 2). For example, ribcage impedance signals measured during normal breathing and airway obstruction look similar, but the pneumotachometer confirms that there is no air flow during airway obstruction. Therefore, breath-detection algorithms based on measurements from a single sensor are likely to be inaccurate.

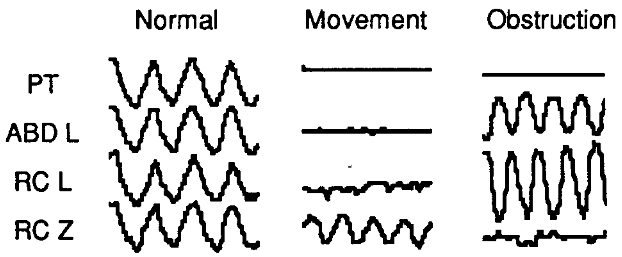

Fig. 2. Recordings of normal breathing, arm movement without breathing, and simulated airway obstruction. PT: pneumotachometer; ABD L: abdomen inductive belt; RC L: ribcage inductive belt; RC Z: ribcage electrodes.

We hypothesize that algorithms based on simultaneous recordings from multiple independent sensors should be able to detect breathing more accurately than algorithms based on the output of a single sensor. We have observed that on healthy adult subjects, ribcage impedance, ribcage inductance and abdomen inductance signals appear qualitatively similar during normal breathing, but differ during abnormal breathing and motion [3], which we attribute to both recording locations and sensing technologies. To test our hypothesis, we have developed breath detection algorithms based on fuzzy logic inferencing [4], which provides a convenient method to combine information from multiple sensing technologies. The remainder of this paper describes our FNN architecture, training algorithm and performance.

2. FUZZY NEURAL NETWORKS (FNN)

Fuzzy inferencing provides a convenient method for using existing knowledge to solve a complex nonlinear system. However, linguistic rules do not necessarily translate to an optimal set of fuzzy rules and membership functions. Adjusting fuzzy system parameters to obtain improved performance can be difficult.

Neural networks (NN) can be trained to perform a nonlinear mapping from input to output space. However, NN are essentially “an unstructured computational black box” [5], since there is no way to embed existing knowledge into a NN, and it is impossible to interpret the trained system.

Horikawa et al. [6] have proposed a FNN architecture, which is a NN that simulates the fuzzy inferencing process. The fuzzy membership functions and rules are determined by connection weights, which allows us to embed existing knowledge into a FNN, and extract knowledge from the trained network.

2.1. FNN Architecture

Figure 3 shows our FNN architecture, which is essentially a Type I network proposed by Horikawa et al. [6], with slight modification. The main differences follow. First, we use a single Gaussian function for each membership function, which can reduce the number of first layer connection weights and nodes by as much as a factor two. Although this may reduce the generality of our membership functions, it simplifies training and reduces the possibility of finding a local rather than global solution. Second, our network input and bias terms are all multiplied by connection weights before the first processing layer, which reduces the number of required layers by one. Finally, we have designed our FNN inferencing and training algorithm for an arbitrary number of input features, each with an arbitrary number of membership functions.

Our network input consists of I features ($x_1, \ldots, x_I$). Each feature defines a universe of discourse, over which we evaluate the membership in an arbitrary number of fuzzy sets. For example, in figure 3 we assume that n, m and p fuzzy sets are defined in the universe of discourse for x_1, x_i and x_I, respectively. Each set is defined by a Gaussian distribution with variance and mean which are related to the connection weights, σl_j and μl_j

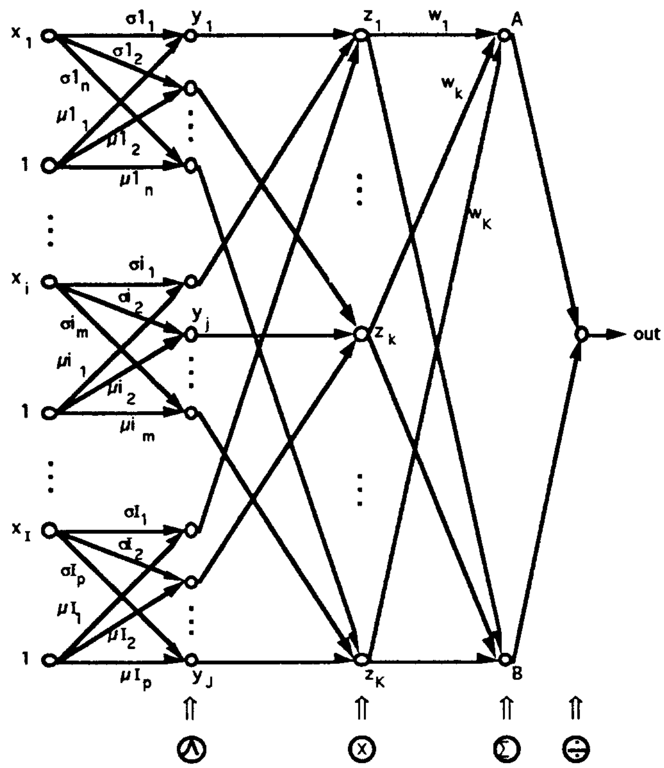

Fig. 3. Fuzzy neural network architecture

The output of the FNN, out, is computed from the input features as follows:

$$y_j = g\left(X_j \sigma_j + \mu_j\right) \tag{1}$$

where

$$g(x) = e^{-(x)^2} \tag{2}$$

$$X = \left[x_1 \cdot \text{ones}(1,n) \quad \ldots \quad x_I \cdot \text{ones}(1,p)\right]^t \tag{3}$$

where ones(1,q) is row vector with q elements, all equal to one,

$$\mu = \left[\mu 1_1 \quad \ldots \quad \mu 1_n \quad \ldots \quad \mu \mathrm{I}_1 \quad \ldots \quad \mu \mathrm{I}_p\right]^t \tag{4}$$

$$\sigma = \left[\sigma 1_1 \quad \ldots \quad \sigma 1_n \quad \ldots \quad \sigma \mathrm{I}_1 \quad \ldots \quad \sigma \mathrm{I}_p\right]^t \tag{5}$$

$$z_k = \prod_{i=1}^{I} y_{V_{i,k}} \tag{6}$$

where $V_{i,k}$ is the entry in the i^{th} row and k^{th} column of V, which contains the index of y which is the i^{th} partial product of z,

$$o = out = \frac{A}{B} = \frac{\sum_{k=1}^{K} z_k w_k}{\sum_{k=1}^{K} z_k} \tag{7}$$

2.2. FNN Training

During training, we adjust the antecedent and consequent parameters w, σ and μ to minimize the output error, E:

$$E = \frac{1}{2}(d-o)^2 \tag{8}$$

where d is the desired output and o is the actual FNN output in eq. (7). By changing the connection weights, we effectively change the membership functions and the consequent of each rule. To change each of these parameters, we perform a gradient descent search.

We adjust the consequent values, w_k, as follows:

$$\Delta w_k = -\eta_w \frac{\partial E}{\partial w_k} = -\eta_w \frac{\partial E}{\partial o}\frac{\partial o}{\partial w_k} \tag{9}$$

where
$$-\frac{\partial E}{\partial o} = (d-o) \tag{10}$$

and
$$\frac{\partial o}{\partial w_k} = \frac{1}{B}\frac{\partial A}{\partial w_k} = \frac{z_k}{B} \tag{11}$$

We adjust the "spread" of each antecedent membership function, σ_j, as follows:

$$\Delta \sigma_j = -\eta_\sigma \frac{\partial E}{\partial \sigma_j} = -\eta_\sigma \frac{\partial E}{\partial o}\frac{\partial o}{\partial \sigma_j} \tag{12}$$

where $-\frac{\partial E}{\partial o}$ is given by eq. (10), and

$$\frac{\partial o}{\partial \sigma_j} = \frac{\partial}{\partial \sigma_j}\left(\frac{A}{B}\right) = \frac{1}{B}\left(\frac{\partial A}{\partial \sigma_j} - o\frac{\partial B}{\partial \sigma_j}\right) \tag{13}$$

$$\frac{\partial A}{\partial \sigma_j} = \frac{\partial}{\partial \sigma_j}\left(\sum_{k=1}^{K} z_k w_k\right) = \sum_{k=1}^{K} w_k \frac{\partial z_k}{\partial \sigma_j} \tag{14}$$

$$\frac{\partial z_k}{\partial \sigma_j} = \begin{cases} 0, & \forall V_{i,k} \neq j, i \in \{1,2,\ldots,I\}; \\ \dfrac{z_k}{y_j}\dfrac{\partial y_j}{\partial \sigma_j}, & otherwise \end{cases} \tag{15}$$

$$\frac{\partial y_j}{\partial \sigma_j} = g'\left(X_j \sigma_j + \mu_j\right) X_j \tag{16}$$

$$\frac{\partial B}{\partial \sigma_j} = \left.\frac{\partial B}{\partial \sigma_j}\right|_{w_k = 1, k \in \{1,2,\ldots,K\}} \tag{17}$$

By increasing σ_j, we decrease the spread of the Gaussian function $g(X_j \sigma_j + \mu_j)$.

We adjust the mean of the membership sets by changing μ_j:

$$\Delta \mu_j = -\eta_\mu \frac{\partial E}{\partial \mu_j} = -\eta_\mu \frac{\partial E}{\partial o}\frac{\partial o}{\partial \mu_j} \tag{18}$$

where
$$\frac{\partial o}{\partial \mu_j} = \frac{1}{X_j}\frac{\partial o}{\partial \sigma_j} \tag{19}$$

Since the values of μ_j must to be negative to obtain a positive mean value for the Gaussian function in eq. (2), by decreasing μ_j, we increase the mean of $g(X_j \sigma_j + \mu_j)$. To always have a membership function with a mean of zero, we do not update the mean for the first membership function for each feature. To always have a membership function with a mean of one, we always update the mean of the last membership function, say μ_r, to a value of $-\sigma_r$.

3. BREATH DETECTION ALGORITHM

Four presumed healthy adult male subjects performed a 15-min experimental protocol which included shallow, normal and deep breathing, arm and leg movements with and without breathing, simulated airway obstruction, yawns, coughs and snores. We simultaneously recorded from a ribcage impedance plethysmograph, ribcage and abdomen inductance plethysmographs and a Fleisch type pneumotachometer (fig. 1). We measured ribcage impedance between Signa II electrodes (Burdick Corp., Milton, WI) placed on opposite midaxillary lines at the level of the nipples. We measured inductance from inductive belts (Ambulatory Monitoring Inc., Ardsley, NY) wrapped around the thorax, such that the ribcage and abdomen belts were centered on the nipples and umbilicus, respectively. We filtered all signals using two-pole bandpass filters with corner frequencies at 0.03 Hz and 10 Hz, and sampled data at 30 Hz using a Macintosh II Computer, National Instruments NB-MIO-16H ADC and LabVIEW software.

3.1. Feature Extraction

We performed template matching to determine possible breathing intervals and computed cross-correlation between sensors during each interval. For each subject, we manually defined a template as all of the samples from one cycle of a normal breath, using an inductance signal which we derived from a linear combination of abdomen and ribcage inductance sensors [2].

We then calculated normalized correlation coefficients between the template and segments of the inductance signal. Sequential segments were all the same length as the template and started every 10 samples. We used the template-matching output to define several contiguous intervals, where each interval was between two positive-going zero-crossings of the template-matching correlation. Therefore, each interval contained one possible breath.

For every interval, we found the maximal value of the template-matching correlation (L_Match) and we calculated the normalized cross-correlation between ribcage impedance and ribcage inductance (L_Z), and between ribcage inductance and abdomen inductance (L_L).

3.2. Initial "Expert System"

We directly use L_Match, L_Z and L_L as features for our FNN. We partition each input dimension using 3 overlapping membership functions with all σ_{ij}=5 and μ_{i1}=0, μ_{i2}=–2.5 and μ_{i3}=–5 for i=1, 2 and 3, corresponding to fuzzy sets Low, Medium and High with means at 0, 0.5 and 1, respectively. We used our hypothesis to generate initial rules of the form: IF L_Match is High and L_Z is High and L_L is high, THEN output is 1, where 0 indicates no breath and 1 indicates breath.

4. RESULTS

Using the pneumotachometer signals, we performed manual scoring to determine if each interval was a breath or not. We compared the output from our fuzzy breath-detection algorithm with our manual scoring to determine the percentage of false-positive (FP) and false-negative (FN) detection. A FP indicates that our algorithm determined a breath had occurred when a breath had not occurred, while a FN indicates that our algorithm determined a breath had not occurred when a breath had occurred.

We tested our algorithm using individual features, and then using all features. We determined algorithm performance before and after training. Table I summarizes the results of using our FNN on 4 subjects, on a total of 944 intervals.

TABLE I. FNN PERFORMANCE (AVE. ± S.D.)

Input Feature	Initial % (FP+FN)	Trained % (FP+FN)	Improvement (%)
L_Match	6.8±3.2	6.9±2.2	–7.4±18.5
L_Z	4.1±0.9	4.5±1.5	–8.3±16.6
L_L	7.3±4.4	3.8±1.0	42.3±16.3
All	5.1±2.4	2.6±0.4	42.9±17.3

5. DISCUSSION

Our breath detection algorithm using a FNN performed well under a variety of conditions, including motion artifact and simulated airway obstruction. The lowest combined percentage of false positive and false negative detections was obtained using the fully connected and trained FNN. We need to test more subjects, including neonates and infants before stronger conclusions can be made.

Our algorithm may be improved in several aspects. First, we could perform additional data acquisition and preprocessing to derive new inputs which may be useful, including the output from adaptive template matching, spectral analysis and other sensors. We could also implement adaptive membership functions and weighting, and improve the defuzzification stage by adding more classes to represent different types of breathing.

ACKNOWLEDGMENT

We would like to thank Adrianis Djohan for his technical support and Burdick Corporation (Milton, WI) for their funding. Kevin Cohen was supported by a graduate fellowship from the Whitaker Foundation.

REFERENCES

[1] AAMI, "Apnea monitoring by means of thoracic impedance pneumography," Assoc. Adv. Med. Instrum., Arlington, VA, TIR No. 4, 1989.
[2] Cohn, M. A., A. S. V. Rao, M. Broudy, S. Birch, H. Watson, N. Atkins, B. Davis, F. D. Stott and M. A. Sackner, "The respiratory inductive plethysmograph: a new non-invasive monitor of respiration," Bull. Europ. Physiopath. Resp., vol. 18, pp. 643–658, 1982.
[3] Cohen, K. P., J. G. Webster, J. Northern, Y. H. Hu and W. J. Tompkins, "Breath detection using fuzzy sets and sensor fusion," Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., pp. 1067-1068, 1994.
[4] Zadeh, L. A., "Fuzzy sets," Information and Control, vol. 8, pp. 338–353, 1965.
[5] Kosko, B., *Neural networks and fuzzy systems*, Simon & Schuster, Englewood Cliffs, NJ, 1992.
[6] Horikawa, S., T. Furuhashi, Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," IEEE Trans. Neural Networks, Vol. 3, pp. 801-806, 1992.