Recurrent Neural Nets for Parameter Estimation

Article · November 2001

Source: CiteSeer

V. PETRIDIS
Aristotle University of Thessaloniki

Athanasios Kehagias
Aristotle University of Thessaloniki

Recurrent Neural Nets for Parameter Estimation

Vas. Petridis and Ath. Kehagias
Dept. of Electrical Engineering, Aristotle University
Thessaloniki, GR 54006, GREECE

Abstract

We introduce a recurrent network architecture to solve a parameter estimation problem; namely we want to estimate the rotor resisatnce of an AC induction motor. A precise estimate of this parameter is very useful for accurate and economical control of the AC induction motor. We propose the *Incremental CRedit Assignment* (ICRA) method for testing online several alternative hypotheses regarding the value of the rotor resistance. These hypotheses are evaluated by a modular recurrent neural network which consists of a number of predictive modules (each tuned to a specific value of rotor resistance) and a decision module. We prove mathematically that maximum credit converges to the "best" parameter value; numerical experiments corroborate our theoretical analysis.

1. Introduction

The operation of the AC induction motor is described (in discrete time) by the following nonlinear state equation .

$$\begin{bmatrix} i_{qs}(t) \\ i_{ds}(t) \\ i_{qr}(t) \\ i_{dr}(t) \end{bmatrix} = \begin{bmatrix} i_{qs}(t-1) \\ i_{ds}(t-1) \\ i_{qr}(t-1) \\ i_{dr}(t-1) \end{bmatrix} + dt \cdot \begin{bmatrix} L_s & 0 & L_0 & 0 \\ 0 & L_s & 0 & L_0 \\ -L_0 & 0 & L_r & 0 \\ 0 & -L_0 & 0 & L_r \end{bmatrix}^{-1} \cdot \left\{ \begin{bmatrix} R_s & 0 & 0 & w(t)L_0 \\ 0 & R_s & w(t)L_0 & 0 \\ 0 & -w(t)L_0 & R_r & w(t)L_0 \\ -w(t)L_0 & 0 & w(t)L_0 & R_r \end{bmatrix} \cdot \begin{bmatrix} i_{qs}(t-1) \\ i_{ds}(t-1) \\ i_{qr}(t-1) \\ i_{dr}(t-1) \end{bmatrix} + \begin{bmatrix} V_{qs}(t) \\ V_{ds}(t) \\ V_{qr}(t) \\ V_{dr}(t) \end{bmatrix} \right\}$$

$$w(t) = w(t-1) + dt \cdot \left\{ \frac{P}{2J} \left(\frac{3P}{4} L_0 \left(i_{qs}(t-1) i_{dr}(t-1) - i_{dqs}(t-1) i_{qr}(t-1) \right) - T_L(t) \right) \right\}$$

Here $i_{qs}(t)$, $i_{ds}(t)$ are stator currents, $i_{qr}(t)$, $i_{dr}(t)$ are rotor currents, w(t) is angular velocity, $V_{qr}(t)$, $V_{dr}(t)$, $V_{qs}(t)$, $V_{ds}(t)$ are input voltages and $T_L(t)$ is input torque. R_s, R_r are stator and rotor inductance, L_s, L_r, L_o are stator, rotor and mutual inductances; J is inertia momentum, P is number of pole pairs of the motor. All of these parameters can be measured directly, with the exception of R_r, which, however, is necessary for the determination of the motor time constant and for efficient angular velocity control. In addition, R_r may vary in time with increasing temperature of the rotor core.

In this paper we present an *Incremental CRedit Assignment* (ICRA) method to solve the problem of R_r estimation. The ICRA method evaluates several possible R_r values and assigns to each one credit according to the predictive power of a corresponding motor model. We present a recurrent, hierarchical, modular neural network implementation of this approach, with a prediction level at the bottom (consisting of a bank of motor models) and a decision level at the top. The decision level is implemented by a recurrent Gaussian neural network which combines the outputs of the motor models. We prove that the credit function converges (with probability one) to the "correct" values, namely, to one for the model with maximum predictive power and to zero for the remaining models.

The idea of combining local models into a large modular network has recently become very popular. It has been used for prediction as well as for classification of both static and dynamic (time series) patterns. A classic exposition of this method appears in [6,7] where the term *local experts* is used in place of our *prediction models*. Our point of view is similar to that of the above papers, insofar we also use local models (predictors) and credit functions. However, ICRA uses structured models, rather than black-box type neural predictors and, in contrast to other approaches, is a *recursive* scheme for *online* credit assignment. This is very appropriate for the rotor resistance problem. We present the *ICRA* method in [1] and a similar, Bayesian-inspired, approach in [2,3]. Similar methods appear in [4,5].

2. Estimation by Classification: Incremental Credit Assignment

We solve the parameter estimation problem by translating it to a *time series classification* problem. This is rather straightforward. We assume that the (observable) stator current sequence $i_{qs}(1)$, $i_{ds}(1)$, $i_{qs}(2)$, $i_{ds}(2)$, ...,$i_{qs}(t)$, $i_{ds}(t)$ originates from a motor with rotor resistance taking one of a finite number of values $R_r(1)$, $R_r(2)$, ... , $R_r(K)$. This results in K distinct motor models, or stator current *classes*; the task then is to classify the observed current to one of the K classes, say the k-th one, and hence estimate rotor resistance as $R_r(k)$.

Hence, in the rest of this section we present the ICRA method as a time series classification scheme. To develop ICRA, start by defining a *decreasing* function of error e:

(1) $$g(e) = e^{-|e|^2/2s^2},$$

which can be implemented by a *Radial Basis Function* (RBF) neuron.. Now we take $e^k(t)$ to be the prediction error of the k-th model at time t, defined by

(2) $$e^k(t) = \left[i_{qs}(t) - i^k_{qs}(t) \quad i_{ds}(t) - i^k_{ds}(t) \right],$$

$i_{qs}(t), i_{ds}(t)$ being the true, observed stator currents and $i^k_{qs}(t)$, $i^k_{ds}(t)$ being the simulated currents computed by k-th model. *We suppose that the errors* $e^k(t)$ *are randomly distributed and follow a Gaussian probability distribution.* Now we take *time varying* quantities $q^k(t)$, k=1,2, .., K, which are credit functions: a high value of $q^k(t)$ implies k-th model has good predictive performance. These quantities evolve in time according to

(3) $$q^k(t) = q^k(t-1) + g\left[g(e^k(t) - \left(\sum_{j=1}^{K} q^j(t-1) g(e^j(t)) \right) \right] q^k(t-1)$$

From eq.(3) we see that the credit fuctions are updated in an incremental manner, similar to a steepest descent procedure, i.e. models with smaller prediction error (hence larger $g(e^k(t))$) are updated with a higher credit. It can also be checked (see [3]) that for all times t we have $\Sigma q^k(t)=1$. At time t the stator current is classified as originating from model k*(t), where k*(t) corresponds to the k-th model with maximum credit function $q^k(t)$. Obviously, the value of k*(t) may change with time, as more observations become available. This concludes the description of the ICRA method, which is described by eqs.(1)-(3). These equations can be implemnted by a simple neural network, illustrated in Fig.1.

Fig.1: Diagram of ICRA Network

The justification of using the ICRA method for resistance estimation is supplied by the following theorem:

Theorem: *Define* $a_k = E[g(e^k(t))]$, $k=1, \ldots, K$. *Suppose* a_m *is the unique maximum among* $a_1, a_2, \ldots, a_K$. *If* $q_m(0) > 0$, *then, as* $t \to \infty$, $q_m(t) \to 1$ *and* $q_k(t) \to 0$ *for* $k \neq m$.

The proof of this theorem is given in [3]; a few comments are in order. Note that $g(e^k(t))$ is a random variable, since it is a function of the error $e^k(t)$. Assuming $e^k(t)$ to be stationary, $a_k = E[g(e^k(t))]$, i.e. the mean value of $g(e^k(t))$, is time independent. Since $g(e)$ is a decreasing function of $|e|$, a large value of a_k implies good predictive performance. In this sense, a_k can be viewed as a prediction quality index and it is natural to consider as optimal the m-th model that has maximum a_m. What the theorem tells us then, is that the $q^m(t)$ associated with m-th model of highest predictive power converges to one, while all other $q^k(t)$'s converge to zero. Therefore the credit functions $q^k(t)$ can be used for succesful classification and hence rotor resistance estimation.

In summary, the ICRA method is based on equations (1)-(3) which can be implemented by a recurrent, hierarchical, modular network. The bottom, prediction level of the hierarchy consists of a bank of predictive motor models, each one implementing a motor model, for a specific value $R_r(k)$. The top, decision level of the hierarchy consists of a module that implements (3); this module can be built from *radial basis function* (RBF) neurons, adders and multipliers, and implemented on a chip.

3. Experiments

In order to evaluate experimentally the performance of the ICRA method, we simulate the AC induction motor, mixing the observation of stator current with additive noise at various noise levels. Each simulation is run for 2500 time steps, each step corresponding to 0.001 seconds of real time; input is a three phase AC voltage of 220 Volts RMS value and torque T_L=5 N·m. The actual motor has the following parameters: R_s=11.58 Ohm, L_s=0.071 Henry, L_r=0.072 Henry, L_o=0.069 Henry, J=0.089 kg·m^2, B=0 Nt·sec/m, P=2; finally we do not take R_r constant, but let it occasionally switch between two values, namely 6.91 and 7.85 Ohms. We use a bank of ten prediction modules (K=10), tuned to R_r values of 1, 2, .. , 10 Ohms. When the actual Rr value is 6.91, the best estimate available is 7 Ohms; when the actual R_r value is 7.85, the best estimate available is 8 Ohms. In Fig.2 we present a characteristic credit profile for the case of noise free observation and in Fig.3 for the case with 10% observation noise.

We evaluate the classification results using the following three indices. First, the number of time steps in which the best R_r estimate is selected is divided by the total number of time steps; this index is called c_1 and its best possible value is 1 (perfect classification). Second, we compute the root mean square error between the actual R_r value and the R_r estimate at every time step; this index is called c_2 and, while ideally its best value would be zero, this is not feasible since the *real* R_r values are not available in our predictive module bank. Third, we compute the root mean square error between the actual Rr value and the best *available* R_r estimate at every time step; this index is called c_3; its best value is zero. These results are presented in Table 1.

c_1 Class.% Correct	c_2 Rr % error	c_3 Rel. Rr % error	Noise Level (% of signal)
0.985	0.020	0.011	00%
0.983	0.020	0.011	01%
0.981	0.021	0.012	03%
0.963	0.022	0.012	05%
0.957	0.022	0.013	10%
0.879	0.030	0.017	20%
0.889	0.030	0.017	33%

Table 1: Resistance Estimation Results

Fig.2:Graph of Credit functions, noise free case. Dashed line is credit of $R_r(8)=8$, Dash-dotted line is credit of $R_r(7)=7$.

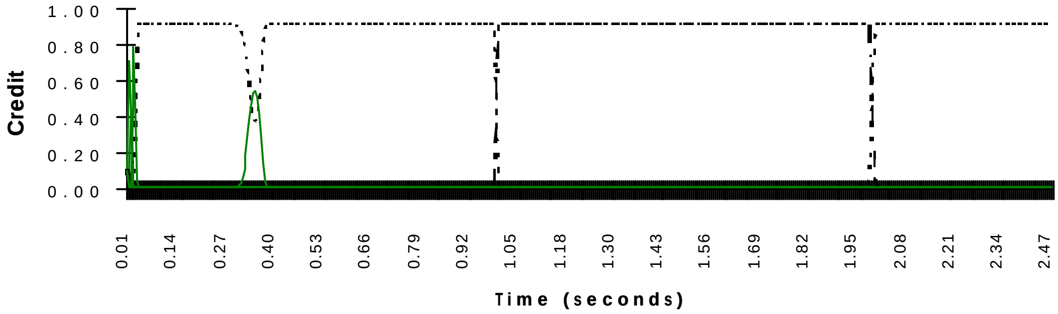
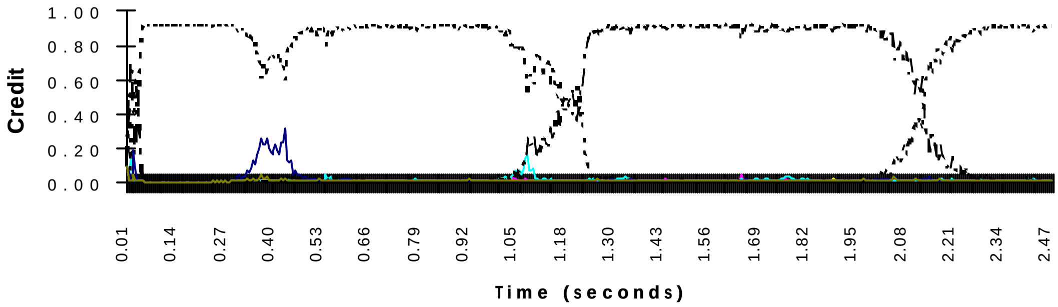


Fig.3:Graph of Credit functions, noise free case. Dashed line is credit of $R_r(8)=8$, Dash-dotted line is credit of $R_r(7)=7$.



We see that performance is very accurate even in the presence of high noise in the observation of current. This, in addition to the online character of the ICRA method and its eacy neural implementation, using RBF neurons, adders and multipliers, make it an attractive method for online estimation of ther AC motor rotor resistance.

References

[1] V. Petridis, and A. Kehagias, A Recurrent Network Implementation of Time Series Classification. *Neural Computation*, Vol.8, No.1, 1996.

[2] V. Petridis, and A. Kehagias, Modular Neural Networks for MAP classifcation of Time Series and the Partition Algorithm. *IEEE Trans. on Neural Networks*, Vol.7 , No.1, January 1996.

[3] A. Kehagias and V. Petridis, Predictive Modular Neural Networks for Time Series Classifcation. To appear in *Neural Networks.*

[4] D.G. Lainiotis, Optimal Adaptive Estimation: Structure and Parameter Adaptation. IEEE Trans. on Automatic Control, Vol. 16, No.2, April 1971.

[5] D.G. Lainiotis and K.N. Plataniotis. Adaptive Dynamic Neural Network Estimation. In *Proc. of IJCNN 1994,* Vol. 6, 1994.
[6] R.A. Jacobs et al., Adaptive mixtures of local experts. *Neural Computation*, Vol. 3, 1991.
[7] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, Vol. 6, 1994.