Copyright © 2002 IFAC
15th Triennial World Congress, Barcelona, Spain

LEARNING OF FIR AND ARX NEURAL NETWORKS WITH EMPIRICAL RISK MINIMIZATION ALGORITHM

Kayvan Najarian

Computer Science Department
University of North Carolina at Charlotte
9201 University City Blvd, Charlotte, NC 28223, U.S.A.
E-mail: knajaria@uncc.edu

Abstract: The probably approximately correct (PAC) learning theory was originally introduced to address static models where the input data were assumed to be i.i.d. In many real applications; however, datasets and systems to be modeled are often dynamic. This encourages the efforts to extend the conventional PAC learning theory to address typical dynamic models such as finite impulse response (FIR) and auto regressive exogenous (ARX) models. This paper presents such extensions for the PAC learning theory and uses the resulting theory to evaluate the learning properties of some families of FIR and ARX neural networks. For ARX models, besides the learning properties of the neural models, stochastic stability of the models are also evaluated.

Keywords: PAC Learning, Nonlinear FIR Models, Nonlinear ARX Models, Neural Networks, Stochastic Stability

1. INTRODUCTION

In a modeling procedure an unknown function "f" is to be estimated to the pre-specified values of accuracy "ε" and statistical confidence "$(1-\delta)$". In order to perform the estimation, based on a set of input-output training data, an approximator function "h" is used to model f. The identification of an unknown system f with a feedforward neural network h can be considered as a typical example of this procedure. The Probably Approximately Correct (PAC) learning theory, proposed in (L.G. Valiant, 1984), deals with the accuracy and confidence of the above-mentioned modeling task. PAC learning and other similar learning schemes allow quantitative evaluation of the learning properties of static modeling procedures in which the data are independently and identically distributed (i.i.d.) in accordance to a probability measure P. However, in many real modeling procedures, the assumption of data being i.i.d. is clearly violated. As indicated in (M.C. Campi and P.R. Kumar, 1996), two important groups of applications to which the results of applicable are "Nonlinear Finite Impulse Response" (NFIR) and "Nonlinear Auto Regressive eXogenous (NARX)" models, where the output depends on the present and the past inputs (as well as the past outputs in the case of NARX). As a result, in such dynamic models, the inouts (as well as outputs) at times "t" and "$t+1$" are correlated and dependent. The importance of FIR and ARX models comes from the fact that in many practical systems, dynamic systems can be efficiently approximated by appropriate FIR or ARX models. The problem of distribution-free learning of linear FIR models trained with the least square algorithm has been addressed by Weyer et al. (E. Weyer, R.C. Williamson and I.M.Y. Mareels, 1996). They use the notion of Vapnik-Chervonenkis dimension to bound the sample complexity of a linear FIR model. One objective of the present paper is to describe the extensions of the convnetional PAC learning theory that allow the assessment of the learning properties of dynamic models. Another objective of the paper is to specify the learning results for general families of neural networks.

The paper is organized as follows: Section 2 describes the basic definitions of the PAC learning theory. The idea of PAC learning with i.i.d. data is extended to PAC learning with m-dependent data in Section 3. The same section gives the learning results of FIR modeling using a general family of neural networks, performed over uniformly distributed input data. In Section 4, the learning theory is extended to learning with α-dependent data that includes learning of ARX neural models. Then the learning results for two families of sigmoid neural networks are given. These results include sufficient conditions for stocachstic stability of neural ARX models. Section 5 gives a detailed discussion of the results of Sections 3 and 4, and is followed by the Conclusions.

2. BASIC DEFINITIONS

In this section, some of the basic concepts of stochastic learning theory, including m-dependency, PAC learning with i.i.d. data, and empirical risk minimization algorithm are reviewed. Suppose X is an arbitrary set. Also suppose that $\mathscr{S}$ and P denote a σ-algebra of subsets of X and a probability measure on $(X,\mathscr{S})$, respectively. A function set $\mathscr{F}$ is defined as a set of measurable functions $f : X \to [-\gamma,\gamma]$. In FIR portion of the paper, we assume that $\gamma = 1/2$. There is nothing special about the interval [-1/2,1/2] and it can be replaced throughout by any bounded interval. Now, consider a modeling task in which the unknown function $f \in \mathscr{F}$ is to be estimated. In order to perform the estimation, a set of training data has to be generated as: $z_n = \left\{(x_i, f(x_i))\right\}_{i=1}^{n}$. Also, assume that each x_i is independently and identically distributed according to the probability measure P. An algorithm A, based on the training data z_n, generates a function $h \in \mathscr{F}$ as an approximator of f. At this point, we can define the concept of PAC learning as follows: Suppose that based on z_n, where $\{x_1,\dots,x_n\}$ are i.i.d. according to the probability P, an approximation task is to be performed as described above. Then a function set $\mathscr{F}$ is said to be PAC learnable iff an algorithm A can be found based on which for any ε and δ, there exists "n" such that:

$$sup_{f\in\mathscr{F}} \mathrm{P}r\{d_P(f,h) \leq \varepsilon\} \geq (1-\delta) \qquad (1)$$

where $d_P(f,h)$ is a distance between f and h defined in terms of the probability P. In this paper and from this point on, we assume that $d_p(f,h) = E_P(|f(x) - h(x)|)$. Another useful concept in function learning is an ε-cover of a function set which is defined as a set of functions $\{g_i\}_{i=1}^{q}$ in $\mathscr{F}$ such that for any function $f \in \mathscr{F}$, there is a g_j where: $d_P(f,g_j) < \varepsilon$. The useful concept to be defined is a specific type of learning algorithm, which is known as the " empirical risk minimization algorithm". Let $\varepsilon > 0$ be specified, and of $\mathscr{F}$ with respect to d_P where d_P is defined above. Then the empirical risk minimization algorithm is as follows: Draw an set of i.i.d samples $\{x_1,\dots,x_n\} \in X^n$, distributed in accordance with P. Define the cost functions: $\hat{J}_i = \frac{1}{n}\sum_{j=1}^{n} |f(x_j) - g_i(x_j)|$, $i = 1,\dots,q$. Now the output of the algorithm is a function $h = g_l$ such that: $\hat{J}_l = \mathrm{m}in_{1\leq i\leq q}\hat{J}_i$.

The last two concepts we define here are m-dependency and α-dependency. A sequence of r.v.s $\left\{Y_i\right\}_{i=1}^{n}$ is said to be m-dependent iff for all j and k, r.v.s Y_j and Y_k are independent if $|j-k| > m$. In other words, in a sequence of m-dependent r.v.s, the radius of dependency is limited to the integer m. Next, the notion of "α-mixing" (also known as strong mixing), is defined that describes a type of stationary random process with exponentially weakening dependency. Let $\mathscr{U}$ and $\mathscr{V}$ be two sub σ-algebras of some σ-algebra $\mathscr{A}$. Then α measure of dependency is defined as:

$$\alpha(\mathscr{U},\mathscr{V}) = \sup_{U\in\mathscr{U},\, V\in\mathscr{V}} \{\, |\mathrm{P}r(U)\mathrm{P}r(V) - \mathrm{P}r(U\cap V)|\} \qquad (2)$$

Now, if $\mathscr{A} = \bigcup_{i=-\infty}^{\infty}\mathscr{Y}_i$, $\mathscr{U} = \mathscr{Y}_{t_0}$, $\mathscr{V} = \mathscr{Y}_{t_0+t}$, and assuming that $\mathscr{Y}_t$ is a stationary process, then $\alpha(\mathscr{Y}_{t_0},\mathscr{Y}_{t_0+t})$ does not depend on t_0 and can be denoted as: $\alpha_Y(t)$. If $\alpha_Y(t)$ approaches zero as $t \to 0$, the process Y is called α-mixing. A more specific case of α-mixing process is geometric α-mixing r.v.s in which $\alpha_Y(t)$ approaches zero geometrically fast, i.e., if there exist:

$$k_1,\, k_2,\, k_3 > 0$$

such that:

$$\alpha_y(t) = k_1 e^{-k_3 t^{k_2}} \qquad (3)$$

then, the process is called geometrically α-mixing.

3. EXTENSION OF PAC LEARNING TO M-DEPENDENT CASES

A fixed-distribution extension of the PAC learning theory to included FIR models was given in (K. Najarian, Guy A. Dumont, and Michael S. Davies, 2001). The general results of this extension were summarized in a thorem that is briefly described here (for more details see: (K. Najarian, Guy A. Dumont, and Michael S. Davies, 2001)):

Theroem 3.1
Assume that there exists a set $\left\{g_i\right\}_{i=1}^{q}$ which forms an $\varepsilon/2$-cover of $\mathscr{F}$. Also, assume that the input training data is a sequence of m-dependent r.v.s marginally distributed according to the uniform distribution. Then, the empirical risk minimization algorithm results in PAC learning of $\mathscr{F}$ with m-dependent training data

$$\sup_{f\in\mathscr{F}} Pr\{d_P(f,h)\le\varepsilon\}\ge(1-\delta) \qquad (4)$$

whenever:

$$n \ge \frac{8(m+1)}{\varepsilon^2} ln \frac{q(m+1)}{\delta} \qquad (5)$$

Notice that similar results can be obtained for other definitions of $d_P(f,h)$ such as the popular $E_P[f(.)-g(.)]^2$. Also, notice that the value q in Theorem 3 is an indication of the complexity of the function set. Although the empirical risk minimization algorithm used in this theorm is ε-dependent and proves PAC learnability to the accuracy of ε, the results can be easily extended to a version of the algorithm which is PAC learnable to any accuracy as mentioned in (M. Vidyasagar, 1997) for i.i.d. cases. Moreover, the parameter q in Inequality (5), plays a vital role in estimation of the overall sample complexity and must be further investigated in the case of modeling with a certain family of neural networks. The key parameter in the calculation of q for a function set is the Lipschitz constant. We now present specialized learning results reported for some families of neural networks.

Suppose that a set of RBFN's $\mathscr{F}$ has members expressed as:

$$f(x)=\sum_{i=1}^{l} a_i\phi_i(r_i)$$

where: $\phi_i(.)$'s are the radial basis functions, l indicates the number of neurons (basis functions), $a=(a_1,\ldots,a_l)$ forms the weight vector of the network with $|a_i|<\infty$ for all i, $r_i=\|x-c_i\|$, where c_i represents the center of the ith basis function. Define:

$$\eta_i=\sup_{x\in[\alpha,\beta]^d}|\tfrac{d\phi_i(r_i)}{dr_i}|$$

and form vector $\eta=(\eta_1,\ldots,\eta_l)$. Further assume that:

$$\sup_{1\le i\le l}\eta_i<\infty\,.$$

and:

$$A=\sup_{\forall a,\eta}\sum_{i=1}^{l}|a_i|\eta_i<\infty$$

Then, the empirical risk minimization algorithm performed over a minimal $\varepsilon/2$-cover results in the PAC learning with m-dependency and the sample complexity of the algorithm can be further specified as ((K. Najarian, Guy A. Dumont, and Michael S. Davies, 2001)):

$$n\ge\frac{8(m+1)}{\varepsilon^2}\left\{\left[\frac{2Ad(\beta-\alpha)}{\varepsilon}\right]^d ln2+ln\frac{(m+1)}{\delta}\right\} \qquad (6)$$

or equivalently:

d

The above results has been even further specialized for the exact forms of basis functions sets. For example, for the families of Gaussian (i.e. $\phi_i(r_i)=\exp(-b_ir_i^2)-\exp(-b_i\|c_i\|^2)$) and Reciprocal Multiquadratic (RMQ) (i.e. $\phi_i(r_i)=\frac{1}{\sqrt{1+b_ir_i^2}}-\frac{1}{\sqrt{1+b_i\|c_i\|^2}}$), the above inequality can be further specified as ((K. Najarian, Guy A. Dumont, and Michael S. Davies, 2001)):

$$n\ge\frac{8(m+1)}{\varepsilon^2}\times \left\{\left[\frac{2\sqrt{2}A_{rbfn}d(\beta-\alpha)}{\varepsilon\sqrt{e}}\right]^d ln2+ln\frac{(m+1)}{\delta}\right\} \qquad (8)$$

and:

$$n\ge\frac{8(m+1)}{\varepsilon^2}\times \left\{\left[\frac{4A_{rbfn}d(\beta-\alpha)}{3\sqrt{3}\,\varepsilon}\right]^d ln2+\ln\frac{(m+1)}{\delta}\right\} \qquad (9)$$

respectively.

The m-dependent PAC learning has also been applied to FIR sigmpid neural networks in (K. Najarian, 2001*c*). A brief discription of these results are given here. Consider a set of feedforward neural networks $\mathscr{F}$ whose members are expressed as ((K. Najarian, 2001*c*)):

$$f(x)=\sum_{i=1}^{l} a_i\sigma(b_ix)$$

where: $0\le\sigma(.)\le 1$ is a smooth sigmoid activation function, l indicates the number of neurons, a_i's are the weights of the output layer and the vector b_i defined as: $b_i=(b_{i1},\ldots,b_{id})$ represents the weights of the first layer. Further assume that:

$$\sum_{i=1}^{l}|a_i|\le\infty\,,\ \sup_{i,j}|b_{ij}|<\infty$$

and:

$$\eta=\sup_{u\in\mathscr{R}}|\tfrac{d\sigma(u)}{du}|<\infty\,.$$

Define:

$$A_{snn}=\sup\sqrt{\sum_{k=1}^{d}\left[\sum_{i=1}^{l}|a_i||b_{ik}|\right]^2}$$

where the above "sup" is taken over the entire parameter space. Then, the empirical risk minimization algorithm performed over a minimal $\varepsilon/2$-cover results in the PAC learning with m-dependency and the sample complexity of the algorithm is given by:

$8(m+1)$

$$\times\left\{\left[\frac{2\eta A_{snn}\sqrt{d}(\beta-\alpha)}{\varepsilon}\right]^{d}\ln 2+\ln\frac{(m+1)}{\delta}\right\} \tag{10}$$

or equivalently:

$$\delta \geq 2^{\left[\frac{2\eta A_{snn}\sqrt{d}(\beta-\alpha)}{\varepsilon}\right]^{d}}(m+1)exp\left[-n\varepsilon^{2}/8(m+1)\right] \tag{11}$$

These results have been even further specified for particular choices of sigmoid functions. For neural networks that apply $\tan^{-1}(.)$ or “atan” (or "tansig") sigmoid functions, the inequality can be written as ((K. Najarian, 2001*c*)):

$$n \geq \frac{8(m+1)}{\varepsilon^{2}} \times\left\{\left[\frac{4A_{snn}\sqrt{d}(\beta-\alpha)}{\pi\varepsilon}\right]^{d}\ln 2+\ln\frac{(m+1)}{\delta}\right\} \tag{12}$$

For neural networks that apply bipolar exponential sigmoid functions of form $\frac{1-e^{-(.)}}{1+e^{-(.)}}$, the inequality can be written as ((K. Najarian, 2001*c*)):

$$n \geq \frac{8(m+1)}{\varepsilon^{2}} \times\left\{\left[\frac{A_{snn}\sqrt{d}(\beta-\alpha)}{\varepsilon}\right]^{d}\ln 2+\ln\frac{(m+1)}{\delta}\right\} \tag{13}$$

The above results are for the input data that are marginally distributed according to a uniform distribution. The new results in the field ((K. Najarian, 2001*a*) and (K. Najarian, 2001*a*)) provide the learning properties of the same families of neural networks assuming an arbitrary distribution. These results are more general and remove the need for the assumption of uniform distribution.

4. EXTENSION OF PAC LEARNING TO α-MIXING CASES

In a dynamic model with feedback from the output together with uncorrelated additive noise, the output of a system is expressed in terms of a function of the the case of a nonlinear ARX (also known as NARX), assuming that $u_{t-q+1}, u_{t-q+2}, \cdots u_{t-d}$ describe the history of the input variable and $y_{t-k}, y_{t-k+1}, \cdots y_{t-1}$ that of the output, then the inpu-output relationship can be described as:

$$y_t = f(y_{t-k}, y_{t-k+1}, \cdots y_{t-1}, \\ u_{t-q+1}, u_{t-q+2}, \cdots u_{t-d}) + \zeta_t$$

where d, $q-d-1$, k and ζ_t represent the degree of the input, the delay from the input to the output, the degree of the output and the additive noise on the system, respectively. Although one can consider multi-dimensional models, here the focus is given to the single-input/single-output (SISO) case. It is also assumed that u_t and ζ_t are uncorrelated sequences of independently and identically distributed (i.i.d.) random variables. The Markov process formed above includes a wide range of dynamic models used in engineering applications such as dynamic neural networks. As a result, all properties of NARX models can be further specified in the case of a particular dynamic neural model. One of the most important properties of a NARX model to be investigated is the stochastic stability of the model introduced in (H.J. Kushner, 1972). The concept of stochastic stability acts as the main key to other issues such as the type of dependency among the data as well as learnability.

The following theorem (in (K. Najarian, 2000)) presents a set of sufficient conditions for stochastic stability and geometric ergodicity of the families of sigmoid neural network discussed above. These conditions are set on the known parameters of the network, and as a result can be easily tested during a practical modeling task. A family of atan sigmoid networks is first considered((K. Najarian, 2000)):

Thereom 4.1
Let:

$$X_t = \\ (y_{t-k}, y_{t-k+1}, \cdots y_{t-1}, u_{t-q+1}, u_{t-q+2}, \cdots u_{t-d},)$$

Take y_t, ζ_t and u_t as defined in (14). Also assume that f is a sigmoid neural network with l neurons of the following general form:

$$f_l(x) = \tfrac{2}{\pi}\textstyle\sum_{i=1}^{l} a_i \tan^{-1}(b_i x)$$

Also assume: $x = X_t$ where: $p = q-d+k$. Further assume that $E[|\zeta_t|] < M_\zeta$ and $E[|u_t|] < M_u$. Define:

$$\omega_j = \sum_{i=1}^{l}\frac{2}{\pi}|a_i||b_{ij}| \tag{14}$$

where $j = 1,\ldots,k$. Suppose: $M_\omega = \max_j \omega$. Also define the following characteristic polynomial: $P(z) = z^k \quad \omega_1 z^{k-1} \qquad \omega$. Then the sequence X_t is geo-

of $P(z)$ is smaller than one. Also, if X_t is stationary then 'y' is geometrically α-mixing.

Having dealt with the conditions for stochastic stability and α-mixing of SNN's, one can move on to the next step which is applying the PAC learning scheme to a task of neural ARX modeling ((K. Najarian, 2000)).

Therem 4.2
Consider a sigmoid neural network as defined above.

Let $\mathscr{F}_l$ be a set of sigmod neural networks with l neurons. Also, define:

$$\bar{n} = \lfloor n \lceil \left(8n/k_3\right)^{1/(k_2+1)} \rceil^{-1} \rfloor \tag{15}$$

For a vector $w = (w_1,\ w_2,\ \dots\ w_d)$, use the notation $|w|_1$ as:

$$|w|_1 = \sum_{j=1}^{d} |w_j|$$

Assume that: $|b_i|_1 \leq \tau_i$ and $\sum_{i=1}^{l} |a_i| \leq C$.

Then, the empirical risk minimization algorithm provides PAC learning with geometrically α-mixing of $\mathscr{F}_l$, i.e. for any ε and δ there exist n such that:

$$\sup_{f \in \mathscr{F}_l} \Pr\{d_P(f,h) \leq \varepsilon\} \geq 1 - (2e(4C+\varepsilon)/\varepsilon)^l \times \left[\prod_{k=1}^{l} (2e(6\tau_k C+\varepsilon)/\varepsilon)^d\right] \times (1+4e^{-2}k_1) \times \exp\left[\frac{-\varepsilon^2 \bar{n}}{64(2+\frac{\varepsilon}{12})}\right]$$

or equivalently:

$$\delta \geq (2e(4C+\varepsilon)/\varepsilon)^l \left[\prod_{k=1}^{l} (2e(6\tau_k C+\varepsilon)/\varepsilon)^d\right] \times (1+4e^{-2}k_1) \exp\left[\frac{-\varepsilon^2 \bar{n}}{64(2+\frac{\varepsilon}{12})}\right] \tag{16}$$

In (K. Najarian, 2000), similar results are provided for ARX models in which exponential sigmoid functions are used (as opposed to $\tan^1(.)$ functions). A brief glance at the results of the above theorem reveals that the introduced bounds are highly conservative, which is the main drawback of many results in the PAC learning scheme. Also, in many real applications, the actual values of k_1, k_2 and k_3 may not be known, and therefore should be estimated using a correlation analysis on the data. As a result, the direct use of the above bounds may fit applications where huge training data sets, along with some information on the

5. DISCUSSION

Now, considering the results of the previous section, the followings remarks can be made:

(1) The presented bounds on sample complexity are sufficient bounds and not necessary ones, i.e. learning might be achievable with fewer number of training points. Therefore, the performance of different models cannot be compared with each other directly from the corresponding bounds on the sample complexities. Nevertheless, the presented bounds on the sample complexities provide us with the order of dependencies between the parameters of the models and their overall learning performance. Moreover, since in order to search for a more complex function, more sample points of the function are needed, the sample complexity can be considered as a measure of structural complexity of the model used for approximation.

(2) The bounds given in the above inequalities indicate that the sample complexity depends not only on the number of neurons and the dimension of the input, but also on the size of the parameter space. This means, even without adding new neurons to the model, and only by increasing the size of the parameter space, one can achieve a more complex neural model. This is the main point a group of researchers including (P. Bartlett, 1996) have been making since early 1990's. They believe that the common trend of adding new neurons to enhance the computational capabilities of neural networks without paying attention to the size of the parameter space may not be the best approach in neural modeling. They recommend that the computational performance of a neural network can be enhanced more systematically (from the point of view of learning theory) by keeping the number of the neurons the same and allowing the parameter space to grow larger.

(3) The direct application of the presented results in a typical task of modeling with a small training data set may not be easy. One approach in using typical results of the learning theory to obtain more reliable models is presented in (K. Najarian, G.A. Dumont and M.S. Davies, 1999), (K. Najarian, 2001*b*). In these papers the learning results obtained for different neural networks have been used to define cost functions that include not only the empirical error but a learning-based complexity term. The existence of the complexity term in the cost function penalizes the learning complexity of the neural model and avoids overfitting the problem when only a small set of data is available. The results of such learning-

complex biomedical system is given in (K. Najarian, G.A. Dumont, and M.S. Davis, 2001).

6. CONCLUSIONS

The learning properties of modeling a dynamic system using an NFIR model are addressed in the case of m-dependent input sequences. Also, the learning theory is extended to another learning theory that addresses ARX modelin with α-mixing data. The sample complexity and other learning results of such modeling procedures for different families of nerual networks (inclusing radial basis function networks and multilayer sigmoid neural networks) have been evaluated. In the case of an FIR model with RBFN's that utilizes a general set of basis functions, the sample complexity is evaluated, and then the results are specialized towards two specific cases, i.e. "Gaussian" and "Reciprocal Multiquadratic" radial basis function networks. In the case of sigmoid neural networks, the learning properties of both FIR and ARX models were evaluated. Moreover, the stochastic stability of such models have been addressed and a set of sufficient conditions for stochastic stability of sigmoid neural networks were given.

7. REFERENCES

L.G. Valiant (1984). A theory of learnable. *Comm. ACM* pp. pp. 1134–1142.

M. Vidyasagar (1997). *A Theory of Learning and Generalization*. Springer.

E. Weyer, R.C. Williamson and I.M.Y. Mareels (1996). Sample complexity of least squares identification of FIR models. *13th IFAC Triennial World Conggrss* pp. 239–243.

H.J. Kushner (1972). *Stochastic Stability, in Lecture Notes in Math.*. Springer, New York.

K. Najarian (2000). *Appliation of learning theory in neural modeling of dynamic systems*. Ph.D. thesis, Dpartment of Electrical and Computer Engineering, University of British Columbia.

K. Najarian (2001*a*). FIR Volterra Kernel Neural Models and PAC Learning. *submitted to Complexity*.

K. Najarian (2001*b*). On Learning and Computational Complexity of FIR Radial Basis Function Networks, Part II: Complexity Measures. *Proc. of ICASSP'2001*.

K. Najarian (2001*c*). On Learning of Sigmoid Neural Networks. *Complexity* **Vol. 6, No. 4**, 39–45.

K. Najarian, G.A. Dumont and M.S. Davies (1999). A learning-theory-based training algorithm for variable-structure dynamic neural modeling. *Proc. Inter. Joint Conf. Neural Networks (IJCNN99)*.

K. Najarian, G.A. Dumont, and M.S. Davis (2001). Modeling of Neuromuscular Blockade System Using Neural Networks. *Proc. of IEEE-*

K. Najarian, Guy A. Dumont, and Michael S. Davies (2001). PAC learning in Nonlinear FIR Models. *International Journal of Adaptive Control and Signal Processing* **15, Issue 1**, 37–52.

M.C. Campi and P.R. Kumar (1996). Learning dynamical systems in a stationary environment. *Proc. 31th IEEE Conf. Decision and Control* **16, no. 2**, 2308–2311.

P. Bartlett (1996). The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *Amer. Statistical Assoc. Math. Soc. Transactions* **17**, 277–364.