Sufficient Conditions for Error Back Flow Convergence in Dynamical Recurrent Neural Networks

Alex Aussem
LIMOS/ ISIMA - University Blaise Pascal, Clermont Ferrand II
B.P. 125, 63173 AUBIERE Cedex FRANCE - E-mail: alex@isima.fr

Abstract

This paper extends previous analysis of the gradient decay to a class of discrete-time fully recurrent networks, called Dynamical Recurrent Neural Networks (DRNN), obtained by modelling synapses as Finite Impulse Response (FIR) filters instead of multiplicative scalars. Using elementary matrix manipulations, we provide an upper bound on the norm of the weight matrix ensuring that the gradient vector, when propagated in a reverse manner in time through the error-propagation network, decays exponentially to zero. This bounds apply to all FIR architecture proposals as well as fixed point recurrent networks, regardless of delay and connectivity. In addition, we show that the computational overhead of the learning algorithm can be reduced drastically by taking advantage of the exponential decay of the gradient.

Keywords - Recurrent neural networks, Gradient descent, Forgetting behavior.

1 Introduction

The use of recurrent neural networks is useful and for some cases inevitable for sequence processing and dynamical system modelling. This paper extends the analysis of the gradient error back flow and the "forgetting behavior" [1, 2, 4, 7, 9] to a fairly general class of fully recurrent architectures obtained by modelling the recurrent synapses as Finite Impulse Response filters (FIR) [1, 15]. The resulting FIR-recurrent structure encompasses a large variety of locally and globally recurrent neural architecture proposals that have emerged in the literature (see for instance [3, 5, 11, 14] for a review) as well as fixed point recurrent networks [12]. By virtue of the recurrency, this versatile model is formally defined as a dynamical system described by a system of nonlinear difference equations, whose internal variables are the hidden unit activations. Therefore, these state-space models are also referred to as Dynamical Recurrent Neural Networks (DRNN) [1].

The architecture being defined in Section 2, the Temporal Recurrent Backprop algorithm is recalled in Section 3, and careful algebraic manipulations are conducted in Section 4 with a view towards formally identifying explicit and interpretation-friendly conditions for the convergence of the error back flow for all neural architectures that are special cases of the DRNN framework, regardless of whether or not they necessitate a relaxation phase to settle down to a fixed equilibrium point - the existence of which is out of the scope of this paper.

2 The DRNN Model

DRNN are delayed networks operating in discrete time in which multiplicative interactions of scalars are substituted by temporal convolution operations, necessitating a time-dependent weight matrix [1, 15]. A general formulation is a coupled differential equations in vector form:

$$\begin{aligned} \mathbf{T}\frac{d\mathbf{v}_k(t)}{dt} &= -\mathbf{v}_k(t) + g(\mathbf{u}_k(t)) + \mathbf{i}_k \\ \mathbf{u}_k(t) &= \sum_{d=1}^{D} \mathbf{W}_k^{d\,T}\mathbf{v}_{k-d} + \mathbf{W}_k^{0\,T}\mathbf{v}_k(t) \end{aligned} \qquad (1)$$

where N is the number of units, $\mathbf{v}_k(t)$ and $\mathbf{u}_k(t)$ are the N-dimensional vector activity and internal inputs of the units at time t, whose values are determined at each step k by the relaxation of the system (1) of coupled differential equations. $\mathbf{i}_k$ denotes the external input vector of dimension N and $\mathbf{T}$ is a diagonal matrix of positive time constant. Let the matrix element w_{ij}^d denote the connection strength, or coupling, from the d-th delay link connecting node i to node j and D the maximum connection delay, i.e., the filter order. The weights for the d-th delay connections heading to unit j at iteration k are specified by the vector $\mathbf{w}_k^d(j) = [w_k^d(1,j), \ldots, w_k^d(N,j)]^T$, and the $N \times N$ matrix $\mathbf{W}_k^d = [\mathbf{w}_k^d(1), \mathbf{w}_k^d(2), \ldots, \mathbf{w}_k^d(N)]$ contains all adaptive weights of delay $d \leq D$ at iteration k. For notational convenience, we won't distinguish between input, hidden, output and bias units in the following formulae. Consistent with our notations, we note $g(\mathbf{u}_k) = [g(u_k^1), \ldots, g(u_k^n)]^T$ under the mild assumption that the nonlinear activation function, $g()$, is $\mathcal{C}^1$ with bounded derivatives.

3 Temporal Recurrent Backprop

The introduction of time delays leads to somewhat more complicated training procedure because the current state of a unit may depend on several events having occurred in the neighboring neurons at different times. Classically, the goal is to find a procedure to adjust the weight components w_{ij}^d so that a given fixed initial conditions $\mathbf{v}_0$ and a set of input vectors $\{\mathbf{i}_k\}$ taken from the training set result in a set of fixed points (the existence of which is assumed), whose components along the output units have a desired set of values $\{\mathbf{d}_k\}$. This will be accomplished by our minimizing a quadratic function $E = \frac{1}{2}\sum_{k=0}^{k_f} \mathbf{e}_k^T \mathbf{e}_k$, which measures the errors $\mathbf{e}_k = \mathbf{d}_k - \mathbf{v}_k$ between the desired and the actual fixed point values along a trajectory or epoch starting at iteration 0 up to k_f. In an epochwise adaptation, a new weight is generated at the end of the trajectory by $\Delta w = -\eta \frac{\partial E}{\partial w}$, where η is the learning rate. The primary goal is to find an expression for the term $\partial E/\partial w$. In order to clarify our notation, let $\mathbf{G}_k$ and $\mathbf{G}'_k$ denote the $N \times N$ diagonal matrixes constructed from $g(u_k(j))$ and $g'(u_k(j))$ respectively, for $j = 1, \ldots, N$. Note that for sigmoid functions, $\mathbf{G'}_k = \beta \mathbf{G}_k(1 - \mathbf{G}_k)$ (Recall that $g'(h) = \beta g(h)(1 - g(h))$).

Theorem 1 - *If the DRNN model is convergent over the trajectory $0 \leq k \leq k_f$, then the gradient of cost can be computed at time step k_f by*

$$\frac{\partial E}{\partial \mathbf{W}^d} = \sum_{k=d}^{k_f} \boldsymbol{\delta}_k \mathbf{v}_{k-d}^T \quad \forall d \leq D \tag{2}$$

where the vector $\boldsymbol{\delta}_k$ is given by $\boldsymbol{\delta}_k = \mathbf{G'}_k \mathbf{y}_k$ and $\mathbf{y}_k$ is the solution of the following backward difference equation:

$$\mathbf{y}_k = [\mathbf{I} - \mathbf{W}_k^0 \mathbf{G'}_k]^{-1} [\mathbf{e}_k + \sum_{j=1}^{D} \mathbf{W}_{k+j}^j \mathbf{G'}_{k+j} \mathbf{y}_{k+j}] \tag{3}$$

with the boundary conditions: $\mathbf{y}_{k_f} = [\mathbf{I} - \mathbf{G'}_{k_f} \mathbf{W}_{k_f}^0]^{-1} \mathbf{e}_{k_f}$ and $\mathbf{y}_j = \mathbf{0}$ for $j > k_f - D$.

The demonstration of this proposition has been given elsewhere [1]. This generalized ∂ rule puts recurrent-backprop [12], temporal backprop [15] and backprop-through-time (BPTT) [13, 16] into a single framework. As BPTT in its general from, the procedure suggested here makes necessary the artifact of unfolding the error-propagation network in a reverse way in time. However, because the storage and memory requirements grow proportionally with k_f, the procedure becomes swiftly inhibitory from an implementation standpoint when long trajectories are to be considered. This is also the major flaw of BPTT. Fortunately, the cost due to the duplication of the hardware, here, can be maintained within acceptable limits, as we will show next.

4 Conditions for Backprop Convergence

Typically, all the individual weight adjustment resulting from the duplication of the connections in the unfolded error-propagation network must be recombined to compute the overall change.

This is achieved by summing the individual gradient components over the overall unfolding interval. Define $\mathbf{y}_k^{k+n}$ as the solution of Eq. 3 at time k when all $\mathbf{e}_{k+j}$ are set to $\mathbf{0}$ except for $j=n$. Let $\boldsymbol{\delta}_k^{k+n} = \mathbf{G'}_k \mathbf{y}_k^{k+n}$, we may now expand the delta-rule (2) by incorporating the terms $\mathbf{y}_k^{k+n}$ and $\boldsymbol{\delta}_k^{k+n}$

$$\Delta \mathbf{W}^d = \sum_{k=d}^{k_f} \Delta \mathbf{W}_k^d = \eta \sum_{k=d}^{k_f} \boldsymbol{\delta}_k \mathbf{v}_{k-d}^T = \eta \sum_{n=d}^{k_f} \sum_{k=d}^{n} \boldsymbol{\delta}_k^n \mathbf{v}_{k-d}^T \tag{4}$$

As we are focusing on a samplewise algorithm, we restrict our attention to the second sum as $n \to \infty$, the convergence of which is *a priori* not guaranteed insofar as the number of layers in the unfolded error-propagation network is increasing with k_f. More explicitly,

Definition 1 - *The gradient based training procedure is said to be convergent at each intermediate time step on the trajectory if* $\lim_{n\to\infty} \sum_{k=d}^{n} \boldsymbol{\delta}_k^n \mathbf{v}_{k-d}^T$ *exists.*

Therefore, for the sum to converge, not only shall $\boldsymbol{\delta}_{k-j}^n$ tend to zero with respect to j but the decay rate must be large enough. With the aim of establishing sufficient conditions for the sum to converge, we now state the fundamental result of this paper:

Theorem 2 - *Let the instantaneous scalar cost be* $E_n = \frac{1}{2}(\mathbf{d}_n - \mathbf{v}_n)^T(\mathbf{d}_n - \mathbf{v}_n)$ *and consider the error vector* $\mathbf{y}_k^{k+n^T} = \frac{\partial E_{k+n}}{\partial \mathbf{v}_k}[\mathbf{I} - \mathbf{G'}_k \mathbf{W}_k^{0^T}]^{-1}$. *Let* $k_f > 0$ *denote the trajectory length of the system and* $\mu = max(g'(h))$. *Let*

$$a_d = \max_k\left(\frac{\mu\|\mathbf{W}_{k+d}^d\|}{1-\mu\|\mathbf{W}_k^0\|}\right), \quad 1 \le d \le D \tag{5}$$

If $\rho = \sum_{d=1}^{D} a_d < 1$ **then** $\|\mathbf{y}_{k_f-(kD+j)}^{k_f}\| < \rho^{k+1}\|\mathbf{y}_{k_f}^{k_f}\|$, $\forall k \ge 0, \forall j = 1,\dots,D$

Proof: The behavior of $\mathbf{y}_k^{k_f}$ can be inferred from the backward recursive equations (3). It suffices to set $\mathbf{e}_j = 0$ for $j = 0, 1, \dots, k_f - 1$. This yields the relation

$$\mathbf{y}_k^{k_f} = [\mathbf{I} - \mathbf{W}_k^0 \mathbf{G'}_k]^{-1} \sum_{d=1}^{D} \mathbf{W}_{k+d}^d \mathbf{G'}_{k+d} \mathbf{y}_{k+d}^{k_f} \tag{6}$$

for $k < k_f$, with the following boundary condition $\mathbf{y}_{k_f}^{k_f} = [\mathbf{I} - \mathbf{W}_k^0 \mathbf{G'}_k]^{-1}\mathbf{e}_{k_f}$. The vectorial equation (6) governs the dynamic of the error terms. Let $\mu = \max(g'(h))$ so that $\|\mathbf{G'}_k\| < \mu, \forall k$. Let us assume that $\|\mathbf{W}_k^0\| < 1/\mu$ for $k = 0 \dots, k_f$ so that the convergence of $\mathbf{I} + \mathbf{W}_k^0 \cdot \mathbf{G'}_k + (\mathbf{W}_k^0 \cdot \mathbf{G'}_k)^2 + \dots$ is assured (Notice at this point that $\|\ \|$ refers to any matrix of vector norm *compatible* with each other, i.e., $\|\mathbf{A}\|_p = \sup_{x\neq 0}\|\mathbf{A}x\|_p/\|x\|_p, p = 1, 2, \infty$). Hence for any iteration $k = 0, \dots, k_f - 1$, we have

$$\begin{aligned}\|\mathbf{y}_k^{k_f}\| &\le \|\mathbf{I} - \mathbf{W}_k^0 \mathbf{G'}_k\|^{-1} \sum_{d=1}^{D} \|\mathbf{W}_{k+d}^d\|\,\|\mathbf{G'}_{k+d}\|\,\|\mathbf{y}_{k+d}^{k_f}\| \\ &\le \frac{\mu}{1-\mu\|\mathbf{W}_k^0\|} \sum_{d=1}^{D} \|\mathbf{W}_{k+d}^d\|\,\|\mathbf{y}_{k+d}^{k_f}\|\end{aligned} \tag{7}$$

Now define $a_d = \max_k\left(\frac{\mu\|\mathbf{W}_{k+d}^d\|}{1-\mu\|\mathbf{W}_k^0\|}\right)$, $1 \le d \le D$. From (6) and (7), we deduce the following inequality

$$\|\mathbf{y}_k^{k_f}\| \le a_1\|\mathbf{y}_{k+1}^{k_f}\| + a_2\|\mathbf{y}_{k+2}^{k_f}\| + \dots + a_D\|\mathbf{y}_{k+D}^{k_f}\| \tag{8}$$

The time dependency of the coefficients has been removed. This allows the analysis of a univariate series, $\{u_n\}$, defined by $u_n = \sum_{j=1}^{D} a_j u_{n-j}$, to be carried out. $\{u_n\}$ is of interest since $u_n \geq ||\mathbf{y}_{k_f-n}^{k_f}||$ for $n = 0, \ldots, k_f$, with the boundary conditions $u_n = ||\mathbf{y}_{k_f-n}^{k_f}||$ for $n = 0, \ldots, D-1$. Since $u_n < \max_{j=1 \ldots D}(u_{n-j}) \cdot \sum_{j=1}^{D} a_j$, and considering that $u_j < (\sum_{j=1}^{D} a_j) \cdot u_0$ for $j > 0$ and $u_0 = ||y_{k_f}^{k_f}||$, it is readily proven by induction that

$$||y_{k_f-(kD+j+1)}^{k_f}|| \leq u_{kD+j+1} < (\sum_{j=1}^{D} a_j)^{k+1} ||y_{k_f}^{k_f}||, \quad \forall k \geq 0, \forall j = 0, \ldots, D-1. \tag{9}$$

Therefore, $\sum_{j=1}^{D} a_j < 1$ ensure the convergence of the back-propagation dynamic, completing the proof.

The theorem says, in essence, that the terms $\mathbf{y}_{k_f-n}^{k_f}$ are bounded above by a series decaying at exponential rate with respect to n provided $\rho < 1$. In regard to the epochwise training, $\rho < 1$ translates into very appealing conditions:

Corollary 1 - *When the weights are kept fixed over the trajectory,*

$$\sum_{d=0}^{D} ||\mathbf{W}^d|| < 1/\mu \tag{10}$$

are sufficient conditions for the convergence of the training procedure.

The above inequality set a penalty on the large weights, regardless of delay and connectivity, to ensure the convergence of the error back flow. Typically, when the network admits fixed-points at each time step, the residual back-propagated errors drop off exponentially through the past time slices, rapidly becoming negligible. The decay rate, ρ, can then be written as:

$$\rho = \sum_{d=1}^{D} \frac{\mu ||\mathbf{W}^d||}{1 - \mu ||\mathbf{W}^0||} \tag{11}$$

More specifically, with compatible norms $||\mathbf{A}||_1 = \max_j \sum_i |w_{ij}|$, $||\mathbf{A}||_\infty = \max_i \sum_j |w_{ij}|$, and $||\mathbf{A}||_F = \left[\sum_{i,j} |w_{ij}|^2\right]^{1/2}$, we obtain three easily-interpreted sufficient conditions to ensure training convergence:

$$\mu \sum_{d=0}^{D} \max_i \left(\sum_j |w_{ij}^d|\right) < 1, \quad \mu \sum_{d=0}^{D} \max_j \left(\sum_i |w_{ij}^d|\right) < 1, \quad \mu \sum_{d=0}^{D} \left[\sum_{i,j} |w_{ij}^d|^2\right]^{1/2} < 1 \tag{12}$$

- **Related results** - The upper bound is in nice agreement with Hochreiter and Schmidthuber 1995's result for standard recurrent networks [17]. For example, if N is the matrix size, they find that setting $\forall i, j, \; |w_{ij}| < w_{max} < 4/N$ and $\beta = 1$ will result in the exponential decay with a decay rate $\tau = N w_{max}/4 < 1$. Clearly, setting $\mathbf{W}_k^0 = 0$ and $D = 1$ transforms the DRNN into a standard recurrent network, i.e., with a single delay fixed to 1. Notice that, for sigmoïd transfer functions, $\mathbf{G'}_k$ is by definition an $N \times N$ diagonal matrix constructed from $g'(h) = \beta g(h)(1 - g(h))$ so that $g'(h) < \beta/4$. Hence $\mu = 1/4$. Consequently, $\rho < 1$ reduces to a somewhat weaker condition $||\mathbf{W}_k^1|| < 4$ for all k ($\beta = 1$), since one or several weights may exceed w_{max}. Conversely, it is readily seen that $||\mathbf{W}_k^1|| < 4$ is satisfied if $|w_{ij}| < 4/n$ since $||\mathbf{W}|| < N w_{max}$. Similarly, Frasconi et al.'s 1992 forgetting behavior condition [6], i.e., $|w_{jj}| < 1/d$ where $d = \max_h (g'(h)) = 4$, for local feedback networks, is also a particular case of Theorem 2.

  Interestingly, the existence of $[\mathbf{I} - \mathbf{G'}_k \mathbf{W}_k^0]^{-1}$, and thereby the *stability* of the fixed points, are also guaranteed. It suffices to set $D = 0$, Theorem 2 yields $||\mathbf{W}_k^0|| \cdot ||\mathbf{G'}_k|| < 1$ which clearly is a sufficient condition for the existence of $[\mathbf{I} - \mathbf{G'}_k \mathbf{W}_k^0]^{-1}$.

- **Tailoring the number of back-propagations** - Now that we have a lower bound for the error back flow exponential decay rate, ρ, we can estimate the true gradient, as accurately as desired, with minimal computational effort as stated below:

  Corollary 2 - *When the weights are kept fixed over the trajectory, it suffices, at iteration n, to allocate memory to accommodate a maximum of sD back-propagations where s is the smallest integer satisfying*

$$s > \frac{1}{\ln\rho} \cdot \ln\left[\frac{\epsilon(1-\rho)(1-\mu||\mathbf{W}^0||)}{\eta\mu N D||\mathbf{e}_n||}\right] \tag{13}$$

  to guarantee that the weights are updated with precision ϵ at each time step.

  The proof is readily obtained from Theorem 1. It is not detailed for conciseness. Formula (13) provides an implementation-friendly stopping-criterion. As far as the memory allocation is concerned, the upper bound for s ensures that the bookkeeping always fit the time interval spanned by the error back flow. It is worth remarking that s varies as a function of $(1-\rho)/\ln\rho$, and thus diverges swiftly for values of $\rho > 0.9$. $s \to \infty$ when $\rho \to 1$, and $s \to 1$ when $\rho \to 0$.

- **Numerical applications** - Usual values such as $N = 10, D = 1, \mu = 1/4, \eta = 10^{-2}, \epsilon = 10^{-6}, ||\mathbf{W}^0|| = 1, \rho = 10^{-1}$, with a single output, yield a sensible value for s equal to 5 by the above formula. In other words, the absolute error on the weight update at each time step during the trajectory is guaranteed to remain below 10^{-6} (i.e., $||\Delta\hat{\mathbf{W}}_n^d(s) - \Delta\mathbf{W}_n^d|| < 10^{-6}$) after only 5 replications of the network reversely in time. Now suppose we increase ρ's value to 0.5, we obtain $s = 17$ and for $\rho = 0.9$ we obtain $s = 121$. As expected, an increase of the the lower bound for the error back flow entails a dramatic increase of s.

- **Link with regularization** - To ensure that the gradient-based training is convergent, we wish to constrain the model to remain in a subset of the functional set by solving

$$w = \arg\min_w E(w) \quad \text{subject to} \quad \sum_{d=0}^{D} ||\mathbf{W}^d|| \leq \frac{1}{\mu} \tag{14}$$

  According to the Khun and Tucker theorem (take a differential norm e.g., Frobenius norm), there is an implicit equivalence between solving (14) and minimizing a modified version of the cost: $w = \arg\min_w \left(E(w) + \xi\sum_{d=0}^{D}||\mathbf{W}^d||\right)$. The second term in the cost function confines the problem to a restricted set, favoring smooth functions by indirect means. It appears as a *weight decay* as it corresponds to an exponential decay of the weights towards zero. The links between regularization and Bayesian inference are well established (see [8] for instance). Assuming a gaussian distribution of the output errors, minimizing the regularized cost (14) is equivalent to maximizing the posterior solution assuming the following pior on the weights:

$$P(w \mid \alpha) \propto \prod_{d=0}^{D} \exp\left(-\alpha||\mathbf{W}^d||\right) \tag{15}$$

  where w is a vector made up from all weights and α is the hyper-parameter. Interestingly, the training convergence and the statistical analysis of recurrent neural networks concur - by very different considerations - in the observation that the small weights should be priviledged. This formal regularization *stabilizes* the training procedure by indirectly *enforcing the decay* of the gradient when it is back-propagated reversely in time through the error-propagation network.

5 Conclusion

This paper provides upper bounds on the norm of the weight matrix to ensure the convergence of the error back flow. The bounds hold for trajectory networks as well as fixed point networks.

References

[1] A. Aussem, F. Murtagh and M. Sarazin, “Dynamical recurrent neural networks - towards environmental time series prediction,” *International Journal of Neural Systems* **6**, No. 2, pp. 145-170, 1995.

[2] A. Aussem, “Dynamical recurrent neural networks towards prediction and modelling of dynamical systems ,” *Neurocomputing* **28**, pp. 207-232, 1999.

[3] P. Baldi, “Gradient descent learning algorithm overview: A general dynamical systems perspective,” *IEEE Trans. on Neural Networks* **6**, pp. 182-195, 1995.

[4] Y. Bengio, P. Simard and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Trans. on Neural Networks* **5**, pp. 157-166, 1994.

[5] P. Campolucci, A. Uncini, F. Piazza, B.D. Rao, “On-line learning algorithms for locally recurrent neural networks,” *IEEE Trans. on Neural Networks* **10**, No. 2, pp. 253-271 1999.

[6] P. Frasconi, M. Gori, et G. Soda, “Local feedback multilayered networks,” *Neural Computation* **4**, pp. 120-130, 1992.

[7] S. Hochreiter and J. Schmidhuber, “Long short term memory,” *Neural Computation* **9**, pp. 1735-1780, 1997.

[8] C. Goutte, *Statistical Learning and Regularisation for Regresion.* Ph.D. Thesis, Université of Paris 6, December, 1997.

[9] T. Lin, B.G. Horne, P. Tino and C.L. Giles, “ Learning long-term dependencies in NARX recurrent neural networks,” *IEEE Trans. on Neural Networks* **7**, pp. 1329, 1996.

[10] M.W. Mak, K.W. Ku and Y.L. Lu, “On the improvement of the real time recurrent learning algorithm for recurrent neural networks,” *Neurocomputing* **24**, pp. 13-36, 1999

[11] S.W. Piche, “Steepest descent algorithms for neural network controllers and filters,” *IEEE Trans. on Neural Networks* **5**, pp. 198-212, 1994.

[12] F.J. Pineda, “Generalization of back-propagation to recurrent neural networks,” *Physical Review Letters* **59**, pp. 2229-2232, 1987.

[13] D.E. Rumelhart, G.E. Hinton and R.J. Williams, “Learning internal representations by error propagation,” in *Parallel Distributed Processing Explorations in the Microstructure of Cognition*, D.E. Rumelhart and J.L. McClelland, Eds., Cambridge, MA: MIT Press, Bradfords Books, vol. 1, pp. 318-362, 1986.

[14] C.T. Tsoi and A.D. Back, “Locally recurrent globally feedforward networks: a critical review of architectures,” *IEEE Trans. on Neural Networks* **5**, pp. 229-239, 1994.

[15] E.A. Wan, *Finite Impulse Response Neural Networks with Applications in Time Series Prediction.* Ph.D. Thesis, Stanford University, November, 1993.

[16] P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in Behavioral Sciences*, Ph.D. thesis, Harvard University, Cambridge, MA, August 1974.

[17] R.J. Williams and D. Zipser, “Experimental analysis of the real-time recurrent learning algorithm,” *Connection Science* **1**, pp. 87-111, 1989.