Available online at www.joac.info

ISSN: 2278-1862

Journal of Applicable Chemistry

2014, 3 (4): 1337-1422

(International Peer Reviewed Journal)

State-of-Art-Review (SAR-Invited)

Mathematical Neural Network (MaNN) Models
Part IV: Recurrent Neural networks (RecNN) in bio-/chemical- tasks

K RamaKrishna[1], V. Anantha Ramam[2] and R. Sambasiva Rao[2*]

1. Dept of Chemistry, Gitam Institute of Science, Gitam University, Visakhapatnam, 530 017, **INDIA**
2. School of Chemistry, Andhra University, Visakhapatnam 530 003, **INDIA**

Email: karipeddirk@gmail.com, rsr.chem@gmail.com

Accepted on 9th July 2014

(Dedicated to Tubati Venkateswara Rao, B O L, asu-kavi, on his nineteenth birth anniversary)

ABSTRACT

Recurrent- (Rec_) NNs contain partial or full backward connections between layers of feed-forward (FF-) NNs. Here, in addition to the present input, past values of inputs/outputs/intermediate states influence the final output. Hopfield NN, proposed in 1982 by John Hopfield, has cyclic architecture and employs Hebbian learning. It brought back life to NN methodologies which were dormant since 1969. Partial recurrent NNs, for instance, Elman with feedback from hidden to input layer and Jordon from output to input layer won laurels to I/O transformation of dynamic systems. NNs with feedback connections have intrinsic dynamic memory that reflects the current output as well as the previous inputs in the transformed space, while feed-forward-NNs (MLP, RBF) do not have previous associated "memory". Rec_NNs models temporal as well as long term behavior in a dynamic system. The typical training algorithms employed in Rec-NNs are Back propagation through time (BPTT), Real-time recurrent learning (RTRL), Atiya-Parlos recurrent learning (APRL), Alopex, long short term memory (LSTM) and extended decoupled Kalman filter (DEKF). Rec-NNs have been extensively used in the data analysis with time series and state-space models with better performance even for systems with more non-linearities. Evolution is a timely modification, invention, discovery in nature. Man desires and achieves the target now and then within in short span through pooling up intelligence of intelligentsia, might of mighty and deep rooted micro-processes of a process from the mother-nature. Rec-NNs not only mimic the popular ARMA, NARMA, NARMAX, Weiner, Hammerstein and Volterra time series models, but also predict more complicated profiles. The impact of Rec_NN approach in medicine- / chemo- / enviro- / dieteto- /quali-metrics rendered modeling of partially understood dynamic systems viable for in depth understanding/control. The key applications are in nuclear power plants, environmental monitoring, weather forecast, greenhouse control, food quality, multi-variate-multi-response calibration, chemometrics, fuel cells, fermentation, ECGs, Schizophrenia, epilepsy, dementia, sleep apnea, HIV, ICU, robots, autopilot mode of aircraft landing, fault detection, communications, linguistics, seismic signal processing etc. The theoretical stability analysis proved the convergence, reliability of this class of NNs. The reported limitations of Elman_NN are a focus for improvements in training algorithms, hybridization and new intermediate means for trial solutions. The solution methods for linear projection equations and quadratic programming tasks are realised with this NN. Rec NNs with IFR and IIR filter characteristics *and those for multiple time series made a mark in engineering tasks. Fuzzy and SOM paradigms are hybridized with rec-NN resulting in Rec-Fuzzy-NN and Rec-SOM-NN. Multiphase processes are modeled by hierarchical Rec-NNs. A few advances in Rec-NN worth mentioning include multi-feed-back-layer-, and higher order recurrent-neuro-fuzzy-NNs. The future prospects of recurrent NNs in chemical, biological sciences, chemo-informatics, and genomics are multifold.*

Keywords: Recurrent _neural_ networks, Elman_Jordan_Hopfield_architectures, Time series, NARMAX, IIR_FIR_filters, Rec_Fuzzy_NNs, Applications, Chemistry, Medicine, Engineering.

Contents: Recurrent Neural Networks in bio-/chemical- tasks

A4 State_Of_the_Knowledge_of_the- (SOK-) Time_Series_Models (TSM)

INTRODUCTION

The experimental, theoretical and simulation science progressed with steadfast growth in enquiring microscopic processes from macroscopic observations and vice versa with the then state-of-art-instruments/knowledge/intuitive power. The approximate solution of exact equation and exact solution of approximate equation have been the brain behind the progress of computational tactics of quantum chemistry/physics. The truncation of Taylor's infinite series to finite order, integration within finite limits of infinite boundaries, termination of infinite algebraic series to finite number of terms based numerical accuracy, tolerance in geometric comparison of diverse patterns/ similar patterns with distinct distortion are all valid computational modules even today under the cover of CPU time, cost/ benefit ratio, information_extracted/ information_unconvered etc. The current methods for these pursuits in nature are a consequence of evolution for at least 13.7 billions of years since the origin of this universe. But, viable/non-viable computational algorithms are not even few centuries of age [1-4 and references therein]. This provoked the enquiry of what are biological/ mathematical/computational procedures evolving with time. The utopian (now impossible with man-made technology) methods in twenty second century will a capsule of best of best of the changing paradigms. In continuation of our studies in Chemometrics, neural networks and swarm intelligence [1-20], the current state of recurrent neural networks and their applications reported in literature are reviewed [21-427]. Auto resonance theory (ART), the brain child of Grossberg, along with its software products are versatile in real time utility in defense and industry in addition to extensive intricate research output in high end applications almost in all disciplines. Grossberg put forwarded non-linear_difference_differential_equations in prediction and learning theories during nineteen sixties [1, 21-24]. In nineteen eighties, Cohen and Grossberg proposed a NN functioning as a stable associative memory. In 1984, Hopfield introduced a continuous time network (now popular after his name) has multi-facet applications in pattern classification, image processing and associative memories. It is a special case of Cohonen-Grossberg NN.

1.1 Limitations of (static or non-recurrent) feed-forward-NNs

Training connection weights of neurons in feed-forward-single/multi- layer perceptron NNs with back propagation of errors algorithm is essentially a steepest descent procedure popular in optimization for non-pathological functions. In these NNs, the data flow is only in the forward direction from input to output through hidden layers. The neuron- and/ or layer wise connections are thus distinct from those of recurrent NNs (vide infra). ARTx series is a realization of classification procedure of auto resonance theory of Grossberg. The three unsupervised modules (ART1, ART2 and fuzzy-ART) have a unique advantage of no need of a priori specification of number of clusters and accepts binary, analogue and floating point input data. Self-organizing scheme of Kohonen is noteworthy in its neighborhood structure and a rectangular basis function. It is a sought after method for visual display of multi-dimensional data in two- or three-dimensions. The life cycle of NNs during approximately last three-quarters-of-a-century period is briefed in Appendix A1.

1.2 Recurrent NNs

In recurrent neural networks, the output depends not only on the current input to the network, but also previous inputs, outputs, or intermediate states (outputs neurons of hidden layer) of the network. A recurrent neural network (Rec-NN) is a class of NNs with feedback connections existing between any two/more or all layers (KB.1). MLP-NNs with tapped delay, recurrent, recirculation connections, and recurrent neurons form Rec-NNs. Rec-NNs with time varying systems are sometimes referred as dynamic recurrent networks.

KB.1(Knowledge base): Recurrent NNs

If	SLP-NN &
	Reverse connections from HL → IL
Then	Elman-NN
If	SLP-NN &
	Reverse connections from OL → IL
Then	Jordon-NN
If	SLP-NN &
	Reverse connections from OL → IL &
	HL → IL &
	self feedback connection
Then	Fully Rec-NN
If	FF-NN &
	Number of hidden layers = 1&
	TFHL = SG &
	No reverse connections
Then	SLP-NN

Input: The input to Rec-NNs can be a single/multiple time series, data of explanatory variable(s) with a functional relationship with response or vector of values exhibiting associative memory. The neural MOORE machine is a popular and general Rec-NN. Elman-Rec-NN is a widely used MOORE machine. A Rec-NN is represented as a sextuple. The output of static neural network is calculated directly from the input through feed forward connections. They do not have feedback connections either between neurons or layers. But, the architecture of Rec_NN contains at least one cycle. In addition, some times, the connections between processing units form a directed cycle/graph. This results in temporary memory and models dynamic spacio-temporal trends.

1.2.1 Partial- (Local-) Rec_NNs

The simplest Rec-NNs were extension of SLPs/MLPs with at least one recurrent connection of a neuron (KB. 2). In a locally recurrent NN with layered architecture, the feedback connections are to the proceeding layer only. The examples include the feedback of the hidden layer to the input (Elman-NN) or output to the hidden layer. However, the layer wise recurrent connection did miracles in I/O transformation. Hopfield-NNs is an example of partially (layer wise) recurrent NNs containing one cycle. A locally-Rec-NN with two hidden layers approximates state-space trajectory simulated by any Lipshitz continuous function with arbitrary accuracy. In locally recurrent NNs, poles can easily be detected and training is faster compared to globally-recurrent-NNs.

Rec_NN(X,U,Y,f,h,x0) as a sextuple

X	:	States of space	f	:	Next-state function
U	:	Input	h	:	Output function
Y	:	Output	x0	:	Initial state of the Rec_NN

Applications_ Local Rec_NNs: They found importance in studying chaos in attractors of Chua's circuit, modeling continuous polymerization and neutralization processes/ signal processing, control of non-linear systems, prediction of speech utterances and fault diagnosis in sugar evaporator/three tank laboratory system.

KB. 2: Local and global RecNNs

If	forward connections &
	feedback connections
Then	RecNN
If	Output of OL is fed back
	to input of IL
Then	Globally recurrent
If	Each neuron in a layer has one or
	more lagged loops around itself
Then	Locally recurrent

1.2.2 Global- Rec_NNs

The feedback connections are previous time step (weighted) outputs. If more than one previous time is necessary, outputs for nlags are considered [280]. In the memory neuron network, each feed forward neuron is associated with a memory neuron. The single scalar output of it summarizes the history of past activation of that unit. The direct feedback of the output layer to the input layer constitutes the globally recurrent NN class viz. Jordan-NN.

Fully connected Rec_NN: In a fully-connected Rec-NN, each unit of the NN is connected to every other unit.

2. Dynamic-/ Recursive-/ Recurrent-NNs

2.1 Biological inspiration: In cellular systems, there are nets of biological neurons connected in a recurrent fashion, which are responsible for oscillating behavior. The parallel neuron pathways of nerves systems, from multiple sensory neurons to the multiple motor nerves [235] play a key role. The pathways compete with each other where in mutual inhibition affect the process. The learning process can be considered as the adjustment of the competitive excitation/inhibition among the parallel paths. Generally, the inhibition strength is increased for a wrong output. During firing activity of biological neurons, persistent oscillations arise due to combination of dynamics between cellular and synaptic micro-processes. The periodic oscillations are observed in heartbeat, respiration, mastication, locomotion and memorization/retrieval. In the cerebral cortex of mammals and brains of insects, oscillations originate from stimuli. The time delays during integration in communication are common. In nonlinear continuous-time RecNNs, output varies with time as a function of its current state and also on the values at previous (lag) time steps. Nonlinear differential equations map the cell signaling processes and thus DOEs are applicable models. The solutions of non-linear DOEs are easily mapable into Rec_NNs. Thus, recurrent NNs are good mathematical models for dynamic systems and bio-modeling of cell processes.

FF_NNs with Backward connections	
BAM	Boltzman machine
Hopfield	
Elman	Jordan
BSB	ARTx

2.2 Chemical inspiration: Reverse connections are common in equilibrium chemistry/ chemical kinetics from simple to complicated reactions. The biochemistry, geo- /environment- /processes also involve cyclic path ways with even acyclic components.

Appendix 3 describes various types of recurrent neurons and their interconnections widely employed in mathematical NN procedures.

2.3 Hopfield NN

Biological inspiration: In biological systems, associative rather than content-addressable style of memory exists. The output of biological cells is a result of a series of spikes of potentials and the frequency vs. total action potential looks like a sigmoid wave.

John Hopfield in 1982 published a short, but clearly presented communication dealing with a fully connected SLP in the forward and reverse directions. This paper gave rebirth to the field of NNs which were dormant from 1969 onwards with the criticism of Minsky and Peppert against perceptron which fails separation of clusters with non-linear boundaries (XOR). Hopfield_NN is a typical instance of both locally and globally recurrent NN [228]. Based on the nature of noise, Hopfield NNs are categorized as deterministic and stochastic ones. Data type classifies them into discrete (binary/bipolar) and continuous varieties. Hybrid Hopfield NNs imbibed the other paradigms.

Architec.Hopfield-NN

The input, Hopfield and output neurons are fully connected in the forward direction. In addition, there are reverse (recurrent) connections from output to input through Hopfield and also from output directly to input neurons. But, there is no self-feed-back connection for any neuron. This architecture forms a cyclic structure unlike in Elman or Jordon NNs. Further, the weights in the forward and reverse directions are symmetric. The transfer function for Hopfield neurons is non-linear basis function which has an inverse. Hebbian learning is used to train Ws. The sought after transfer function, sigmoid is used in Hopfield layer. The convergence of dynamics of Hopfield NN is guaranteed. Sometimes, Rec-NNs behave chaotically [235].

Applications.Hopfield

Hopfield-NN is used for robust content-addressable memory, storing examples (memory), analog to digital conversion, telecommunication in military/civil domains, pattern recognition and optimization tasks. This NN stores/recalls information, performs error correction and searches for nearest correct values. Typical applications of Hopfield NN are torque minimization of redundant manipulators, constrained and unconstrained optimization , shortest path for internetwork routing, support vector machine training, graph coloring, image processing and control systems design. Hardware implementation of Hopfield NN is another unique feature.

Table 1: Classification of cervical cells with Hopfield NN

Class		% of correct classification	
0	**1**	**Tr**	**Te**
Normal	Severe	100	97
Normal	Mild	68	66
Moderate	Severe	70	67

Cervical cells classification:

In this decade, growth of biological knowledge of cancer in animal surgery/experiments and monitoring human patients through clinical and surgical protocols is exponential. Around 500 genes were implicated and list is now longer. At the moment there are around hundred US_FDA approved drugs. But, there is no remarkable decrease in death rate even in countries with latest therapies/diagnosis/ monitoring/surgical expertise available. This demands more peer studies of genes and a new angle why naked mole rat is resistant to cancer [429], while 90% mice die of carcinoma. The classification of cervical cells **(Table 1)** with five features shows the potentiality of Hopfield NN.

Associative memories: Hopfield NN functions as an associative memory also. For, example abbreviation and acronym pairs can be stored. Later, one can retrieve acronym for an abbreviation or vice versa.

The advantages and limitations of Hopfield NNs are briefed in chart 1.

Chart 1: Hopfield-NN

+ It has convergent activation dynamics
+ Hopfield-NN >> MLP in learning

— If prototypes are not available,
Then classical Hopfield NN is not suitable

— Storage capacity of Hopfield NN is 0.15*n
(n : number of neurons)

— If elements stored > limit
Then retrieval/recall becomes problematic
▶ Remedy : Brouwer method

— Converges to local minima

Theoritical analysis_Hopfield

Hu [314] derived sufficient conditions for uniform boundedness of high-order-Hopfield-NNs with time varying delays using Lyapunov functional. Xu [201] analysed linear stability of a six-neuron binary associative memory- (BAM-) NN with discrete delays. It exhibited Hopf bifurcation. Takagi-Sugeno (T-S) Fuzzy model consists of a set of linear sub model [292]. Lie studied mean squares exponential stability of stochastic – Fuzzy – Hopfield NN with discrete and distributed time varying delays. Wang [346] reported a robust decentralized controller which a shows global asymptotic stability in probability of the uncertain stochastic environment.

Recent advances in Hopfield NN

Quaternionic neuron_Hopfield NN

Hopfield-Rec-NN with quaternion neuron was proposed recently. The TF is step function and is same for each component in quaternion function.

Chaotic Hopfield NN

Zheng [311] reported the study of 3-neuron chaotic Hopfield NN with Lyapunov exponents spectrum, bifurcation diagram, power spectrum and topological Horseshoe theory. This NN exhibits novel double scroll chaotic attractors. Wang investigated delayed_ higher_order_Hopfield_NN with unbounded TFs.

The sufficient conditions for the existence of periodic solution are derived. Qiu [325] found conditions for periodic solution of higher_order_Hopfield_NN using Lyapunov method and linear matrix inequality techniques. Kaslik reported the theoretical analysis of discrete-time-delayed-Hopfiled NN of p-neurons with ring structures. The existence of Fold/Cusp, Neimerk-Sacker and Flip bifurcation is proved. Further, a theoretical proof is given for the occurrence of Marotto's chaotic behavior. Liu [199] reported sufficient conditions ensuring equilibrium number, local stable number, global stability and complete stability of Hopfield NNs with multilevel activation functions. Hopfield NNs are extended [197] in developing dynamic resource allocation algorithms with advantages like distribution of resources and different connections over same resources. It is applied in scheduling and optimization tasks.

2.4 Elman-NN

In 1990, Elman proposed a hidden_layer-wise_recurrent-SLP, still a popular neural model for dynamic systems and time series data.

Architec. Elman-NN: Elman NN has architecture as that of SLP with an additional feedback loop of hidden layer with a single delay. The output of hidden layer is called internal state. The feedback layer is referred as context layer and is represented below the input layer with a connection to the input of the hidden layer (Fig. 1). The weights of recurrent connections from hidden layer to the context layer are fixed at 1.0. These fixed back connections result in the context units always maintaining a copy of the previous values of the hidden units or in other words have a memory unlike FF_NNs. It is a partially recurrent architecture retaining layered configuration. It is also referred as locally recurrent, but globally feed forward NN. Yet, the capacity to retain the history of past information is limited. It is trained with BP algorithm. Hyperbolic, tangent, sigmoid and log sigmoid are used as transfer functions in the hidden and output layers in later versions of Elman model. But, generally purelin TF is preferred in the output layer.

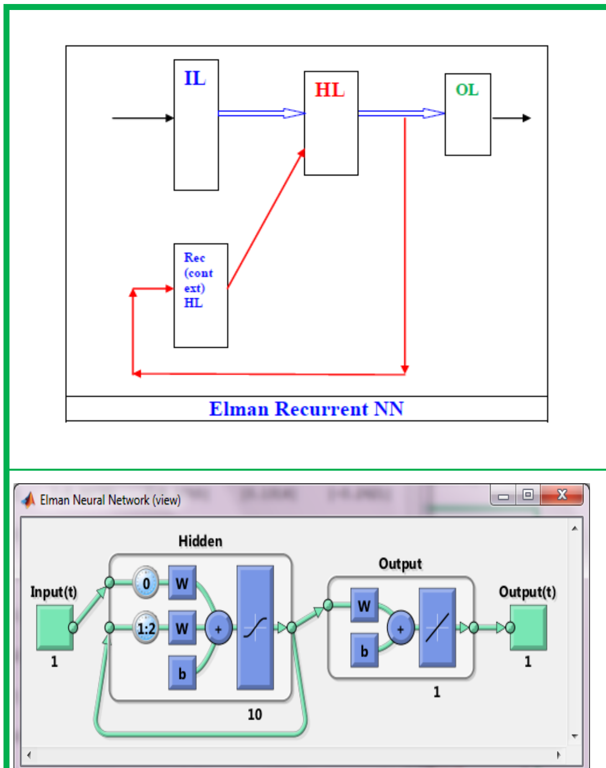


Fig. 1(a): Elman RecNN representation in MATLAB-NN-toolbox

2D- contours are more informative than 3D-surfaces except visual appreciation. Figurative representation of NNs also does not give more information than mathematical/object orientation notation. Keeping this view, we reported Matlab m-files for Elman, Jordan, RBF, MLP and SLP architectures

Functioning of Elman-NN

To start with, the context units are set to zero. The input is propagated like in a standard feed-forward fashion followed by application of a learning rule. Thus, the net effect is activation of external inputs only at the time is equal to zero. The output of hidden units is fed back at every subsequent time steps (t+1, t+2 etc.) to the context layer. The information in the context units is thus a combination of the previous raw input data as well as that coming out of hidden units. Now, both context units and external input units activate the neurons of hidden layer. The output of hidden units is then fed forward to activate the output units. The hidden units then activate the output units as well as context units at the t+2 and so on. This sequence of operations is repeated at all subsequent time steps. In a nutshell, the context units provide information to the network recurrent in time.

Refinement of weights and biases: MSE (mean square error) is the object function and the weight up gradation is through gradient descent with an adoptive learning rate. MATLAB NN tool-box provides a set of learning m-files. At each time step, the error is back propagated to find gradients of error for each weight and bias. For every each epoch, the entire input sequence is presented and the error sequence is generated.

A pre-state (context) layer is a copy of activities in the state layer at the previous time step. Leung and Chan [144] used dual-EKF (Extended Kalman filter) to train Elman-RecNN and to prune ineffective connection weights after the learning phase. Here, one KF estimates the state of the system. The weights are estimated by Recursive_Least Squares. The error covariance matrix of the RLS algorithm is instrumental to eliminate insignificant weights or weak connections between neurons. The simulation studies endorse the present joint-learning–pruning method is effective even for online learning. Pham and Karaboga [156] used GAs to train the Elman and Jordan networks. Recently, Ohta and Gunjib [235] implemented four types of learning mechanisms viz. WTA, negative reinforcement, depression and potentiation and pre-synaptic inhibition are used during the training. The winner takes all (WTA) philosophy is used for the neurons of the state and output layer of Elman_RecNN. The stability and plasticity of the proposed model is studied.

Elman, Jordan and their hybrid Rec_NN

architecture	
Layer	Yes
confluence	[dot product]

Neuron connections	
Within a layer	NO
BeyondSuccessiveLayers	NO
selfFeedBack	NO
Forward	Yes
Forward.Full	Yes

Feed_Foward				
#Layer	Name	#neurons		TF
1	IL	dimx	=1	Identity
2	HL	dimH	60	Kernel
3	OL	dimy	1	PureLin
			Np =999	

Feed_forward		
$Layer_i$	$Layer_j$	W
1	2	WIH
2	3	WHO

Recurrent

	Connections. Feed_back. recurrent	
	From	To
Elman	Out_HL,	Context Layer (CL)
Jordan	Out_OL,	CL

	$Layer_i$	Contex $Layer_j$	W_Rec
Elman	2	2	WRecH
Jordan	3	2	WRecO

	$Layer_i$	CL	W_Rec
Elman + Jordan	2	2	WRecH
	3	2	WRecO

Fig. 1(b): Output of Object oriented m-program (Fig.1(c)) of Recurrent NN architectures

Simultaneous refinement of architecture and Ws of Elman_RecNN:
Subrahmanya and shin [83] proposed a combination of PSO and covariance matrices adaptation based evolutionary strategy to simultaneously refine the architecture and Ws of Elman type RecNNS (Chart 2). Ge et al. [102] proposed dissimilation-PSO to simultaneously evolve architecture, train weights, initial inputs of the context units and self-feedback coefficient of Elman-RecNN. It is applied to speed identification and a controller of Ultrasonic Motors.

Chart 2: simultaneous optimization of NN architecture and Weights

DPSO Discrete PSO	Refinement of structure of a Rec_NN
IPSO Improved PSO	Parameter refinement of each of Structures present in the DPSO

Applications.Elman

Elman_RecNN performs modeling of dynamic systems and sequence-prediction. This is beyond the scope of standard feed-forward-MLP-NNs. The utility of this NN is wide spread in science, engineering, medicine and commerce. The typical applications include monitoring influencing environmental variables [106],

```
%
% oo_ElmJorHyb.m  (R S Ra0) 14-9-10
%
clean

% Network Name
  NN.name ='ElmJorHyb'

 %  --Architecture ----

 ElmJorHyb.connections.Forward=  '[Full,Forward]'

 % confluence
 ElmJorHyb.confluence = '[dot product]'

% Defaults
 Neuron.connections.withinLayer = 'NO'
 Neuron.connections.BeyondSuccessiveLayers = 'No'
 Neuron.connections.selfFeedBack = 'NO'

 architecture.Layer ='Yes'
 Elman = 0; Jordan = 0; Hybrid = 1

% Layer Names

 ElmJorHyb.Layer(1,:).name = 'IL'
 ElmJorHyb.Layer(2,:).name = 'HL'
 ElmJorHyb.Layer(3,:).name = 'OL'

 % Number of patterns
 dimx = 1; dimy = 1; dimH = 60; NP = 11;
 ElmJorHyb.NP = NP
 % Layer dimensions --Data structure

 ElmJorHyb.Layer(1,1).neurons = dimx
 ElmJorHyb.Layer(2,1).neurons = dimH
 ElmJorHyb.Layer(3,1).neurons = dimy

 % Layer TFs

 ElmJorHyb.Layer(1,:).TF = 'Identity'
 ElmJorHyb.Layer(2,:).TF = 'Kernel'
 ElmJorHyb.Layer(3,:).TF = 'PureLin'

 % Weight

 ElmJorHyb.weights(:,1,2) =  'WIH  '
 ElmJorHyb.weights(:,2,3)  = 'WHO  '

 % Weight_recurrent

 if Elman
 ElmJorHyb.weights(:,2,2)  = 'WRecH'
 ElmJorHyb.connections.Recurrent.feedBack=  '[Out_HL, Context]'
 end

 if Jordan
```

```
 ElmJorHyb.weights(:,3,2)   = 'WRecO'
 ElmJorHyb.connections.Recurrent.feedBack=  '[Out_OL, Context]'
 end

 if Hybrid
 ElmJorHyb.weights(:,2,2)   = 'WRecH'
 ElmJorHyb.weights(:,3,2)   = 'WRecO'
 ElmJorHyb.connections.Recurrent.feedBack=  …
'[Out_HL, Context; Out_OL, Context]'
 end
```

Fig. 1(c): Matlab function for Object oriented architectures of Rec_NNs

cabbage growth Chinese fields, sequential controller for automated manufacturing system [153], inverse kinematics solution of robot manipulator with six-degrees of freedom [135], linear_ordinary differential equations (ODEs) [120], cultural heritage, identification of no-cavitation and developing/ super/ jet cavitation of water inside an orifice [33]. Brudzewski [162] used Elman_RecNN in recognition of gas pulses with a small array of semiconductor oxide sensors. The results of table 2 warrant more intensive investigations of systems for foolproof knowledge. A large number of applications in diverse disciplines are described in under the head applications of recurrent_NNs in section 6.

Skin-care: Wan [66] predicted skin-care (tone, spots etc.) with Elman**,** cascade-forward-BP and FF- BP NNs. The key features are selected from questionnaires from women in Beijang.

Microarray data: The simulated non-stationary processes and the real biological microarray continuous time series datasets are studied and the pedagogical rule extraction method imparts explanation rule capability. Ao and Palade [63] introduced an ensemble of Elman-RecNN with SVM.

Chemical industry: A sub discipline of industrial chemical process includes time series (TS) of key production yields and process parameters which help to monitor the health of ongoing schedule and to raise alarming warnings in case of shortcomings/failures. Thissen, Buydens et al. [147] modeled the data of simulated ARMA (4,1), Mackey–Glass, differential pressure in a spinning process from filtration unit with ARMA, SVM, Elman RecNN. A dynamic NN model for predictive control of a distillation column was proposed. The output of the hidden neurons is fed back to the input through one or several time delay units. A sequential orthogonal training is used where in the hidden neurons are added one by one to avoid over training sunspot, exchange rate, McCay-Glass datasets.

Seismic signal: Tiira [161] compared SLP, MLP and Elman NNs in detecting onset of seismic signal using vertical channel data recorded in central Finland. The inputs are four different STA/LTA values computed in seven frequency bands. The training data base was obtained from P-wave signals of 193 teleseismic events and the output was high at onset while a low value for noise. The results are 25% better for detection and reduced false alarms to 50% compared to Murdock–Hutt detector, a popular approach in this discipline. The order of performance is SLP > MLP > [Elman, Jordan] NNs. Baddari et al. [75] studied seismic data inversion with FF_NN, Elman using back-propagation conjugate gradient training algorithms. Djarfour et al. [109] reported seismic data filtering with ElmanRecNN.

Lynx data: Aladag et al. [92] proposed a hybrid Elman-RecNN and ARIMA to model both linear and non-linear components of TS data. The accuracy of forecast of Canadian Lynx data is noteworthy.

Long term prediction in TS: Darus and Al-Khafaji [58] modeled dynamic response and one-step-ahead-prediction of flexible plate excited by sinusoidal force using MLP_BP, Elman-RecNN and ANFIS. Menezes and Barreto [108] studied chaotic laser and a variable bit rate (VBR) video traffic time series datasets with NARX-NN with excellent results compared to time delay NN or Elman-RecNN. The task here is long-term (multi-step-ahead) prediction of univariate time series.

Chaotic systems: Al-Assaf et al. [141] reported NNs for identification of parameters in fractional order chaotic systems. The first step is arriving at features by discrete Fourier transform (DFT), power spectral density (PSD) and wavelet transform. The next step that follows is training with Elman and Jordan RecNNs and prediction of fractional chaotic system parameters are of adequate accuracy. The features of fractional order chaotic systems – extracted from PSD using Welch functions, Multi-resolution (Multi-Resol) wavelets transform (WT) and DFT are the input to Rec-NN to predict parameters of fractional chaotic system. Here, PSD is preferable to the other two transform techniques. Residual analysis using hybrid Elman–NARX neural network along with embedding theorem is used to analyze and predict chaotic time series.

Datasets.simulated_and_real_life: Ardalani-Farsa [81] analysed sunspot, Lorenz, McCay-Glass TS datasets with hybrid Elman-NARX-embedding theorem NN. The results with this algorithm are more accurate compared to those of other dynamic_NNs.

Software failure: Ho et al. [142] trained software-failure data along with the corresponding parameters with MLP, Elman, Jordan NNs and parametric models. Earlier, nonhomogeneous Poisson process models were usually employed.

Detection of oil spills: Ziemke reported [168] the detection of oil spills in sea coasts using segmentation of Doppler radar images from backscatter signals with Elman type NNs.

Industrial operation: Javed et al. [30] reported summation Wavelet Extreme Learning Machine (SW-ELM) excels Levenberg_Marquardt (LM) in batch training of SLP and Elman NNs for industrial datasets viz. pump, industrial dryer, turbofan etc. Portillo et al. [91] detected degradation of the process with varying thicknesses (50 to 100 mm) of wire electrical discharge machining of work piece using Rec_NNs with model validation and test success exceeding 85% and 75%. The prediction of anti-germ performance and ingredient levels detergents [74] are reported.

Portfolio selection: Lin et al. [124] reported Elman is better than vector auto regression (VAR) in dynamic portfolio selection.

Table 2: Precarious results of comparing efficiency of Elman-NN with other techniques

Elman NN is Inferior	**Task**	**Ref**
NARX > Elman	Dementia	**36**
FF-NN (SLP, MLP)	Concrete (Ready mixed delivery system)	**129**
Sub_ connection_NN >> [Elman's, Jordan]	Simulations	**171**
MLP, >> [elman, Jordan, Gen_FF_NN]	Fuel cells performance -- polybenzimidazole-polymer electrolyte membrane	**95**
Elman performance degraded with length of task	Maze learning	**85**
GRec_NN >> [Elman, MLP, [SAA, linreg]	Technical profiles of novice CAD trainees	**57**
RBF>> [Elman, MLP, Hopfiled]	24hour weather forecast at southern Saskatchewan, Canada	**130**
ARMA, SVM >> Elman	Spinning process	
NARX >>Elman	Learning- & demonstrator- robot	**73**
MLP >> [Elman, RBF]	Simulation of evaporation process Of Various climatologic regimes	**50**
MLP>> [Elman, RBF]	Predict sleep apnea’s	**89**
Elman >> MLP_BP probabilistic	Isolated speech recognition	**88**

- RecNNs are proven imbibing and emulating paradigm for dynamic systems.
- If a method is inferior to other (proven better) methods, it warrants a retrospection viz. version of method, Training algorithm, data structure, accuracy/precision, appropriateness etc.

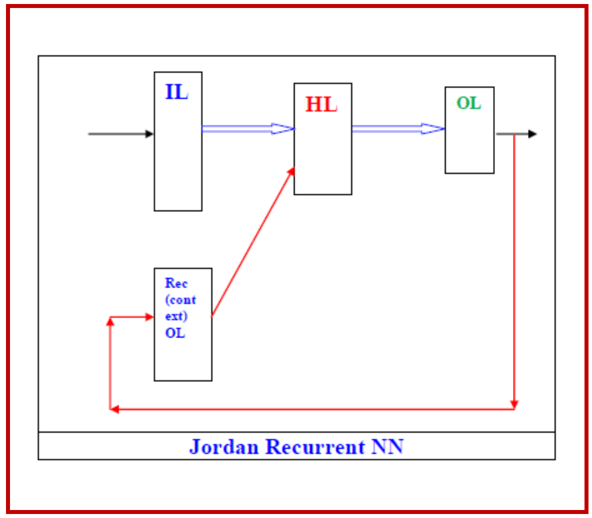
Linguistic analysis: Liou [196] performed and categorization of Shakespeare's writings using trained Elman-NN. The semantic meaning of the words is acquired through an automatic acquisition process.

2.5 Jordan Neural network

Jordan proposed a partially recurrent NN in 1986.

Architec. Jordan-NN: Jordan-Rec-NN is a single hidden layer FF-NN with an additional feedback connection from output of output layer to the context layer shown below the input layer. The context layer is directly connected to the input of the hidden layer with a single delay. Thus, the architecture is similar to the Elman network in the sense that there is feedback loop only from one of the layers of network.

The context units (of Elman or Jordan) contained the information of the history of input and also the effect of transformations occurring in the hidden layers. The behavior of Rec-NN model can be deemed to be the result of culmination of static FF-NN and that in a transformed input space.



Applications. Jordan-NN

Environment

Carcano et al. [174] modeled daily stream flow data in two small catchments with irregular and torrential regimes with Jordan NN. The results are used for reconstruction of drought periods, which focuses management and control of water resources. Darsono and Labadie [181] employed Jordan_NN for control of simulated real-time control of combined sewer system in King County, Seattle.

Method	% classi fication
Trigrams	**92**
Jordan Rec_NN	88
Morphology /suffix	85

Software failure analysis

The critical points of attention in software development are test-case development, failure detection, fixing bugs etc. Nonhomogeneous Poisson process (NHPP) models are popular in tracking software failure task. But, the non-adherence of the assumption of these models is the limitation to arrive at reliable output for real time test environment. Ho et al. [142] studied the effects of different feedback weights in Elman RecNN for software failure using historical data of break-downs. The results are compared with FF-NN, Jordan recurrent model, and traditional parametric growth for software reliability models.

Trinity-College- Printed-Catalogue- Dataset.Jordan-RecNN: The motivation for developing this model was for its potential use in the on-line version of a Trinity College 1872 Printed Catalogue [183], a library catalogue which has entries in 14 different languages spanning over 5 centuries. It was thought that neural networks would perform well where entries to be analysed comprised only a few words. The results of trigrams, morphology/suffix analysis and Jordan RecNN are compared.

Linguistic analysis: It is applied in linguistic analysis and for sequential tasks.

Some typical input and output variables of test cases using Jordan NN are described in Table 2b.

Table 2b: Input and output variables for typical tasks using Jordan_Rec-NNs

NN	Task	Input	Output
Jordan	Thermal processing of	→ Processing time	○ Temperature of the cold

5-8-9-1	canned foods Forecast of one step ahead	→ Temp (retort's and cold point) → Current time ti, & ti−1, ti−2.	point
Jordan Elman MLP	Processes in fuel cells	→ Conditioning temp → Operating temp → Current density	Potential, cathode resistance Ohmic resistance
Jordan Elman SLP MLP Rec_NN	Earth quakes	→ Vertical channel data	Seismic signal onsets

Sub-connection (Sub-connect.) RecNN

Shimohara [171] et al. proposed sub-connection neural network (SCNN). It has feedback-to-weight connections and used for event-driven temporal sequence processing. SC-NN performed better than Jordan and Elman networks for event-driven temporal sequence processes like permutation, combination and integration.

2.6 Elman + Jordan NN

A hybrid of Elman and Jordan NNs consists of a 3-layer NN (SLP) with backward connections from the output of the hidden (second) as well as output of output layer to the context layer (Fig. 2) [343].

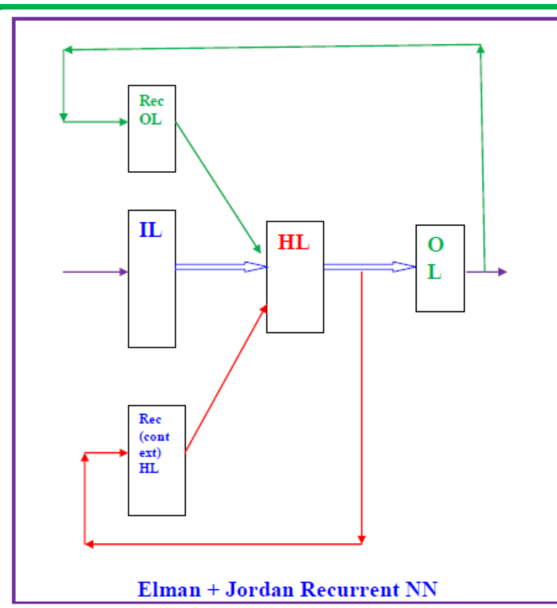


Fig. 2: Hybrid_Elman_Jordan RecNN

Memristor-based Rec_NN

The noise, leave aside its' characteristics, is associated with all (real) life processes, for example, central nervous system (CNS) of biological species and large-scale integration (VLSI) circuits of industrial importance. This is modeled in the framework of memristor-based Rec_NN with stochastic flavor.

Directed acyclic graphs (DAG)-Rec-NN: Baldi [229] reported recursive NNs based on underlying directed acyclic graph (DAG) (Fig. 3), a weight sharing approach. Here, a deterministic parameterized relation is used. The translation-invariant and regular structure of DAG allows reusing the same network at different locations in the graph and protein structure is prediction.

Limitations of Rec_NNs: Simple Rec_NNs fail at vanishing gradients in problems with data having short term lags. Long short term memory (LSTM) (Chart 7) Rec_NN surmounts this shortcoming

Relationship with other techniques: There is correspondence between neural networks and block stochastic models.

3. Training algorithms for Rec-NNs

In training Ws of Rec-NN for time series data, the network is unfolded into MLP, growing layer wise with time. The choice of training algorithms, number of neurons in the hidden layers, activation functions, stiffness of ODEs, structure of W-matrix in Rec_NNs is critical as far as solution of mathematical methods are concerned. Thus, leaning (or training) procedures are newer ones or modified methods used for supervised feed forward neural networks (FF-NNs). The data flow is in the reverse direction complicates the jargon. Back propagation through time (BPTT), maximum likelihood estimation (MLE) and real-time recurrent learning (RTRL), are extensively used to train Ws of Rec_NNs. The common feature of these algorithms is the calculation of derivatives of error sum of squares (ESS) with respect to Ws. Atiya-Parlos recurrent learning (APRL), Alopex, long short term memory (LSTM) and extended decoupled Kalman filter (DEKF) are another set of typical methods employed for training. However, with all these modifications, the training of W in recurrent neural networks is generally very slow.

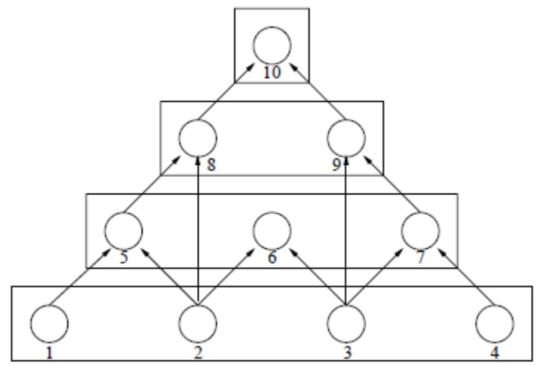


Fig. 3: DAG_RecNNwith Four layers from bottom to top, 10 nodes ; 1-4: input ; 6,10: sink nodes; (Courtesy from Ref 229)

3.1 Back propagation through time (BPTT)

The long term dependencies i.e. modeling relationship between inputs and outputs much earlier in time are not possible through gradient descent methods. But, back propagation through time (BPTT) is an adopted version of standard BP algorithm (Alg. 1). In BPTT, computing the derivatives of error with respect to weights in a recurrent network is reduced to computing the derivatives in each layer of in a feed forward network and adding them in reverse order of forward propagation. The ordered derivatives are appropriately distributed using the chain rule from a given node to all nodes and weights that connect it in the forward direction. CG algorithm can be used in batch version of BPTT. Later, efficient implementations of BPTT are proposed

Alg. 1: *BPTT* for training of Ws of Rec-NNs

Step : 1 *Forward pass*: All inputs pass through the NN in the forward direction
The outputs of hidden layers and output layer are saved in a stack

Step : 2 *Backward pass*: The errors (residuals) are computed at the output layer.
They are back propagated through time in the layers of NNs

Step : 3 *Refinement of weights*: Weights are updated with accumulated updated values.
At each time moment, a feature error is calculated
It is back propagated further through time

If	Jacobean at each time step has all its Eigen values inside the unit circle
Then	Jx(T,n) decreases exponentially

+ With well selected truncated depth, BPTT produces accurate derivatives with reduced complexity and computational time compared to real-time recurrent learning.

+ BPTT can be viewed as unfolding a recurrent network from a time evolving architecture into its multilayer counterpart, there by translating time into space [260]

− BPTT algorithm is computationally very intensive
Remedy : Truncated BPTT

Functioning of BPTT: All the three modes training viz. epoch wise, continuous/real time and their combination are used in BPTT and the topology grows by one layer at every time step. It unfolds Rec_NN into multi-layer FF_NN whenever sequences of patterns are learnt. It involves two phases. The first one is unfolding the NN in time and in the second phase the error is back-propagated through the unfolded network. BPTT integrates backwards in time after the network takes a single step forward. The effects of the number of forward and backward integration steps in training are discussed in literature. The relevant history of input data and network state is saved only for a fixed number of time steps, defined as the truncation depth.

Vanishing gradients: The portion of wC is insignificant for lower values of time resulting gradient values very low. This behavior is called the problem of vanishing gradients or forgetting behavior circumvented in modified BPTT (Chart 3).

3.2 Truncated BPTT

Instead of continuously updating Ws, it is performed at every preset number of steps. For this task, the input data as well as the state of NNs are saved. The information older than truncation depth (depth_trunc) is overwritten, as it is does not have added value in training. Truncated-BPTT algorithm (Alg. 2) still follows the true gradient closely with a reduced computation time.

Alg.2: *Truncated-BPTT*

- The local gradient for jth neuron is calculated
- Back propagation is done upto n-h+1
- Upgrade W

If h is chosen to equal h0
Then BPTT (h; h0) algorithm
reduces to batch wise BPTT

BPTT for NARX-NN

The modified BPTT is employed to train NARX-NN. The entire network is unfolded at the recurrent connections, which appear as jump-ahead connections. They provide a shorter path for back propagating the error through the network. After an epoch (presenting all data) of training, the error is back propagated through the unfolded network path. In the output units of the recurrent network, the local error is computed. It is added to the back-propagated value from the subsequent input unit. The advantages of modified BPTT are in Chart 3.

Chart 3: BPTT.Mod

+ Mod.BPTT diminishes the problem of the vanishing gradients

+ NARX-NN learns long-term dependencies in the data.
[**Reason:** This is due to the fact that the error in the present time step is reduced while taking into account the errors made in the future steps]

3.3 Real-time recurrent learning (RTRL)

RTRL refines Ws of Rec-NNs in real time and thus the title goes for this training algorithm. RTRL is a gradient based algorithm wherein Ws of Rec_NN are refined by minimizing MSE between desired output and observed output at the current time step (Chart 4). Unlike BPTT, this algorithm is local in time but not local in space. It is more of theoretical interest.

Alg. 3: *ALOPEX*

- Output of ith neuron at iteration (iter)
- Update wij
- Calculate probability pij(n)

Chart 4: Limitations and remedial measures in *RTRL*

+ The gradient information obtained from RTRL is employed in gradient descent (GD), conjugate GD (CGD) EKF to update Ws

- RTRL is computationally expensive with the time complexity $O(N^4)$; [N : number of neurons]

- A loose coupling of RTRL with GA did not improve performance
    - Remedy: hybrid RTRL-GA or Restricted RTRL-GA
        — High computational cost at each iteration

- Fails for systems with ten step time lags
  Remedy: LSTM

Stability-RTRL: The weights of Rec_NN are refined by gradient of instantaneous error [231]. The structural/asymptotic/exponential/absolute stability of Rec_NNs under variation of parameters was studied. For the hyperbolic equilibrium points linearization methods are employed. In the case of non-hyperbolic equilibrium points, Lyapunov linear matrix inequations solved the task.

+ Prevention of error signals from decaying quickly as they flow back in time
+ CECs keep track of error as it flows back in time thus, errors elsewhere vanish exponentially

- learning long-term dependencies
  - Remedy:
  - Architecture: segmented-memory recurrent neural network (SMRec_NN)
  - Tr : extended RTRL
    - CPU_time increases with network size
    - If extended RTRL impractical
      - **Remedy :** extended BPTT + layer-local unsupervised pre-training procedure

      + improved learning in SMRec_NN
- Traditional RL fails in cases where long term memory of old sensory inputs is required for immediate prediction.
  Remedy: RL with LSTM

Advances in RTRL: The improvements of RTRL include sub-grouping strategy, constrained RTRL and conjugate gradient (CG) method. The relationship between learning rate and the slope of TF for a class of Rec_NNs trained by RTRL are derived. It reduces the degrees of freedom and thus it is of lower computational complexity. A similar relationship for FF-NNs trained by BP reported.

Blanco et al. [222] proposed a modified version of RTRL with the time complexity $O(N^4L^2)$]. An autonomous learning algorithm for fully connected Rec_NNs trained with RTRL was proposed later. This version does not require the dynamics of the system, but tracks the dynamic behavior starting with initial conditions at zero time. This algorithm was successfully implemented for single_input_single_output-(SISO) second order/linear processes, time-varying/non-linear data, multi_input_ multi _output- (MIMO) time series predictions. Further, this algorithm is successful even in presence of uncertainties. A hybridization of RTRL with APRL increases robustness (Chart 5).

3.4 Alopex

The biological inspiration of Alopex is from human vision. This algorithm is a stochastic parallel process in arriving at the global minimum of an error surface. This is a self-starting algorithm exhibiting global generalization. It transmits global cost function to all the neurons synchronously. This correlation between the global error change and weight change is calculated using a probability index to march forward in the right direction. The individual weights are refined (Alg.3). It is applied for control of autonomous underwater vehicles with Rec_NN architecture (2-10-1).

Alopex outperformed four-layered FF-NN (2-20-10-1). The positive features are described in chart 6. It is similar to SAA in an implicit manner. Unlike in BP and other algorithms, Alopex calculates output error after synchronous change of all weights

Chart 5: Hybrid RTRL

- **APRL**
  + Efficient alternative to RTRL
  + Good generalization
- **RTRL**
  - Less sensitive to transients
- **[APRL + RTRL] Loose coupling**
  + Increased Robustness

3.5 Atiya-Parlos Recurrent learning (APRL)

It is a new online continuous time learning algorithm [232] which proceeds in non-gradient search directions. It minimizes standard quadratic error. The results of APRL with RTRL for Mackey-Glass data are compared. In this instance, the combination of APRL with RTRL was proposed to exploit the speed and robustness of the components. Atiya and Parlos considered discrete NNs and used search directions instead of gradient to minimize the standard error. APRL strategy was extended to continuous time NNs.

In this approach the states are considered as control variables and weights are upgraded to achieve the convergence.

APRL as special case: APRL is a truncated one step backward propagation of instantaneous error which is combined with a momentum term. Also, RTRL and BPTT can be derived from a constrained optimization task where in the quadratic error with constraints reflecting the dynamics of NN is the object function. Mackey-Glass data is analysed using the inputs [y(k), y(k-6), y(k-12), y(k-18)] to predict y(k+84) using APRL training of Rec-NN.

Chart 6: alopex

- + Escapes from local minima very fast
- + Learns by relating patterns of recent history with weight change trends
- + It decreases the network error without explicit computation
- + Does not make any assumption regarding architecture, Objfn or TF
- + Amenable for parallel implementation in hardware
- + Rejects noise

3.6 Long short term memory (LSTM)

LSTM is an Rec_NN surmounting limitations of vanishing gradients. Thus, it models use long delays and a mixture of low and high frequency component signals. However, LSTM-Rec_NN learns the trends after training for more than a thousand discrete time steps. It is used as a powerful function approximator in reinforcement learning in partially observable environment. In a standard LSTM, each unit has its own ActFn instruction. LSTM uses constanterror carousel (CEC), a memory cell containing self-connected linear unit enforcing a constant error flow [225]. An input gate learns to protect CECs from irrelevant input and an output gate learns to turn off cell block (generating irrelevant output). A forget gate allows CECs to reset themselves to zero when necessary. Thus, by monitoring the process on long time in CECs, LSTM is able to bridge time lags (>1000 discrete time steps) between relevant events. A meta-learner using LSTM is a fast leaning procedure for non-trivial classes of functions. LSTM is combined with DEKF in training Rec_NNs. Second_ order_ Rec-NNs trained by LSTM_general are better than single layer NNs as the dimension of the dynamic NN is reduced. But, the neurons have identical set of operating instructions for both activation and learning for generalized_LSTM_Rec NN. The advantages of LSTM and the benefit of hybridization are in Chart 7.

Chart 7: LSTM and its hybrid

FF-NNs

- − Traditional NNs have a problem of gradients
  Remedy: LSTM

LSTM

- + No vanishing gradients hurdle
- − No direct connections between CEC to gates
  Remedy: Peephole connections

LSTM + peephole

- + Fusion technology

Peephole connections

Perez-Ortiz et al. [225] proposed new set of weighted connection links with each CEC to the corresponding gates (Alg. 4). The combination of LSTM with peephole connections is an efficient fusion of modified BPTT and a customized version of RTRL. LSTM-gen if used to LSTM architecture with peephole connections the result is an advantage of an additional source of BP error.

Rec-NN.Hou

Hou et al. [257] used this NN for hierarchical control task of a large scale inter connected dynamic systems.

Architec: It consists of local optimization sub networks and co-ordination sub network. This hierarchical NN uses goal-co-ordination approaches. The constraints in the dynamic equations are treated in a nesting manner. It found that the present NN solves three interconnected sub systems efficiently. Interestingly, it is a globally stable NN.

Alg. 4: Recurrent Network

```
For   memory block = 1 to n
      activation of input gate
      activation of forget gate
      activation of output gate
end
Gradient of y(t) wrt each weight
```

3.7 Decoupled Extended Kalman Filter (DEKF)

Extended Kalman Filter maximizes the posterior probability of variables rather than minimizing the training error. A non-linear autoregressive model of time series with noise both in the process and response was adopted. For estimation of Ws and hidden states EKF algorithm was used. There are no external inputs and recurrent connection and pruning was not taken in to account. The basis of decoupled_EKF is that learning is treated as a filtering problem. The optimum weights are estimated in a recursive manner. Decoupled EKF is tested on simulated (SISO) and experimental datasets (sun spot activity, exchange rate prediction and Mackey-Glass series). It uses joint learning and pruning procedure for online-Rec-NNs. A global EKF to estimate W and hidden state of Rec_NN to maximize the posterior probability rather than minimizing training error are used. Chart 8 points out positive features and limitations.

Chart 8: DEKF

+ Apt for online/ batch training
+ Tractability of computationally hard problem
+ Not local in time or space
+ Fewer training steps
+ Generalization > gradient descent

- Slower in training time series data
- Considers only local inter dependencies

3.8 DEKF + LSTM

On line learning and context sensitive language learning are affected with a combination of DEKF and LSTM [225] procedures.

3.9 Evolutionary Learning: Baldwinian learning cannot be better than Lemarckian in evolving NNs. This is true especially when a large number of weights are involved. The limitation of cellular GAs is a large CPU time in training Rec_NNs. But, an addition of a local learning mechanism to cellar GAs result in a hybrid optimization algorithm. GA is used to train Elman and Jordan RecNNs [156] and nonlinear_dynamic_ system_identification. Differential evolution (DE) PSO and hybrid of DE and PSO are also used [244,251]. Cooperative co-evolution training of Rec_NN is described in Alg. 5.

Alg. 5: Cooperative coevolution.Tr.RecNN 14/131

choose encoding scheme

Neuron or Synapse level

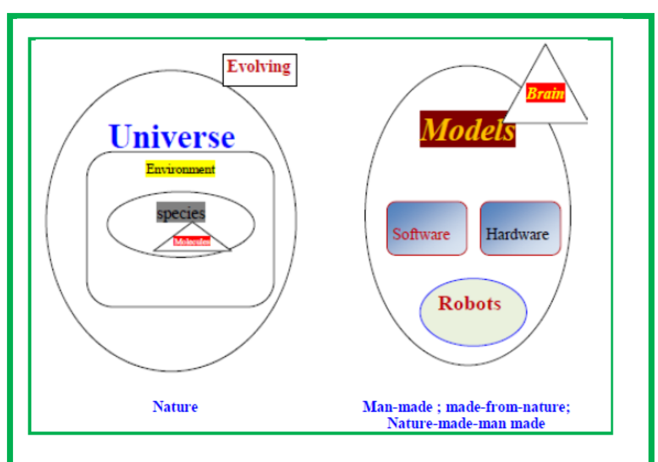
Decompose the problem into k subcomponents

Initialize

cooperatively evaluate each sub-population

```
While   until termination
        for   each Subpopulation
              for   nGenerations
                    Select and create new offspring
                    Cooperatively evaluate new offspring
                    Update sub-population
              end for
        end for
end while
```

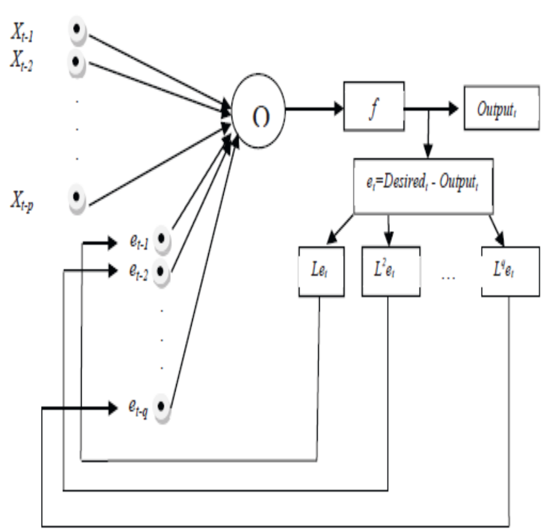
4. Evolution of Rec.NN



Evolution in nature (Fig. 4) is to combat the odds for sustenance of a species for searching food/mate, protecting eggs/off-spring from becoming a prey, suitable habitat, and defense against predators all through the life span. In man_made_computational paradigm (Fig. 4), evolutionary progress is primarily need based to tackle unsolved riddles, circumventing the limitations of the existing methods, avoiding failure conditions, extending scope of applications to more complicated datasets/tasks, enhancing positive features, realizing intuition and reaping the synergism in functioning and robustness. The universal approximation properties of fuzzy system, NNs and NARMAX were proved. Time series modeling was practiced in wide range applications in the last half a century. Linear, non-linear models with parameters, soft techniques, SVM, NNs, evolutionary computation and recently even nature inspired methods like honeybee mating, ant colony optimization (ACO), particle swarm optimization (PSO). Global approach considers the entire system at once, while multiple model paradigms tackle through portioning means. Still one riddle that remains is within how many iterations, for what range of initial values, which training algorithms, fuzzy systems reaches optimum configuration achieving the universal approximation axiom. Tsoi and Back [163] reviewed a unified approach of architectures of discrete time Rec_NNs.

Fig. 4: Evolving modeling, brain and the Universe

4.1 Rec-(multiplicative-neuron)-SLP-NN: It has only one neuron in hidden layer (HL). Auto-regression (AR) as well as moving average (MA) terms is included in the model. The error (yobs-ynn) is fed back and PSO is used for training Ws of NN (Fig. 5).



L(t)	:	Backshift operator
err(t)	:	Error for time t (yobs –ynn)
TF(.) or ActFn(.)	:	Sigmoid(.)
X(t)	:	Observation

Algorithm of Rec-(multiplicative-neuron)-SLP-NN

For i=1:NP
  Cal activation value of neuron
  Cal output of Rec-Mul-Nue.SLP
  Err(i) = yobs-ynn
End for

MSE	:	mean square error
MAPE	:	mean absolute percentage error

arima	:	Autoregressive Integrated Moving Average
Wmes	:	Winter's Multiplicative Exponential Smoothing

Test Data	SARIMA	WMES	MLPANN	RBF-ANN	E-ANN	MNMANN	MS-ANN	L&NL-ANN	RMNMANN
MSE	92.6387	50.4980	13.9891	106.4797	13.4821	40.2006	9.1010	12.7263	8.6289
MAPE	0.2336	0.2204	0.0995	0.3248	0.0990	0.1822	0.0887	0.0944	0.0761
Tr : 124 ; Test : 10 points									

Fig. 5: Algorithm and pollution monitoring analysis with Rec-(multiplicative-neuron)-SLP-NN

Dataset [CO_2,SO_2]. Rec-(multiplicative-neuron)-SLP-NN: The contents CO_2 and SO_2 were monitored monthly during the period March 1995 and April 2006in Ankara, capital of Turkey. The profile of XO_2 with time exhibits a trend and also 12-month seasonal component. The results are compared with SARIMA, WMES, MLP-NN, RBF-NN, E-ANN, MNM-ANN, MSANN, and L&NL-NNs (Fig. 5).

4.2 Sub_connection neural network: It has feedback-to-weight connections. The learning algorithms viz. permutation, combination, and integration are event-driven temporal sequence processes. The results of simulations reveal sub_connection neural network (SCNN) performs better than Elman and Jordan RecNNs.

4.3 Dynamically constructed feedback fuzzy neural controller: This NN has a rule-layer and input-/output- layers. BP algorithm is used in training Ws. The rules are aggregated as and when developed. The pruning procedure of rules and input variables involves detecting irrelevant ones. The deletion is in adherence with optimum accuracy. The advantages of the proposed model are briefed in Chart 9.

Chart 9: Dynamically constructed feedback fuzzy neural controller Tr_alg: Mod-BP)

+ Small sized rule base + SISO NN-architecture for neuro-fuzzy FIS + NN model similar to Lin- ARX + Decomposition of multivariable control rules into three sets of one dimensional rules for each input + Direct relationship between a fuzzy control and a conventional control + Easy connection between fuzzy parameters and operation of controller	Input/out fuzzy sets Number of fuzzy rules Inference methods De-fuzzification methods **Fuzzy set models** ANFIS Sugeno-type fuzzy Differentiable t–norm Differentiable mfs	+ Created from scratch + Simplified evolution of rule base + No need to specify ○ Initial network ○ Architecture ○ Initial membership ○ Functions ○ Initial weights

A multi-dimensional fuzzy-relation rule system can be decomposed into SIMO type systems. The output of these A_1_to_C_1 rule base is inaccurate set of A_m_to_C_n (A: antecedent, C: consequent) rules. In spite of the fact there is decrease of accuracy, A_1_to_C_1 fuzzy rule system has advantages. This is applied to I/O modeling of non-linear system dynamics.

4.4 Decomposed neuro-fuzzy ARX model: It is proposed for a priori unknown dynamic systems with finite input–output. The structure is decomposed into linear AR models (ARX). The rule bases consist of simple rules with a single antecedent and consequent clauses.

4.5 Reservoir computing paradigm in Rec-NN: A large but fixed recurrent part of NN is used as a reservoir of dynamic features. The training is only on the output layer to extract desired information. Just like regularization techniques, it increases generalization. The pruning of some of connections of the reservoir to the output layer is beneficial.

4.6 Convex-hull.RecNN: The numbers of neurons in the first and second layers are equal to the number of data points. In first layer, there is no self-feedback connection. In the second layer there is self-excitation connection for every neuron like in Max-net. The ith neuron of the first layer is connected to the ith neuron of second layer in forward and reverse directions. There are four subnets which are structurally identical but with different activation functions. The ith neuron in the second layer of every network is connected to the ith neuron in the third layer. Each neuron in the third layer is connected to every other neuron and the weights are set to zero. Thus every neuron in the third layer has four connections.

Data processing in convex-hull: In the first layer angles are calculated. Similarly, four subnets in the second layer compute minimum angle. In each iteration, the angles with respect to another neuron are calculated in the first layer. They are passed to second layer. In the second layer, the winner (cliff neuron) is selected and marked as hull-neuron. The weights from the mother to child neurons are set to one. At the end of training the third layer outputs the exact connect hull.

Datasets. RecNN-convex-hull: This NN model is tested on several sets of points in 2-dimensional space.

4.7 Max-Net.Rec-NN: Max-Net is a recurrent-NN with N number of neurons. It is sometimes referred as a single layer NN.

Arch. Max-Net: Each neuron is connected to every other both in forward as well as in reverse directions. Further, there is a self-feed-back connection to every neuron (Chart 10).

Data flow in Arch. Max-Net: During, the iterations (self-organization) of NN, each neuron receive inhibitory input from other nodes through inter-layer (lateral) connections. The outputs of the neurons are updated in parallel. The neuron with maximum value subsequently becomes a winner neuron. The activations of all other neurons are set to zero.

4.8 Hodkin-Huxley neuron-Rec.NN: Rec-NN with stochastic spiking neurons and trained based on reinforcement learning of temporal effect. It is shown that NN with Hodkin-Huxley neurons with dynamic synaptic kinetics can learn appropriate timing of each neuronal firing. The information in NNs is coded by characteristics of timing of each neuron firing, order of firing and relative phase difference of firing.

4.9 Multiple-class-random-NN: It consists of N neurons and receives exogenous and endogenous signals. The excitatory or inhibitory signals are fixed by the neurons to others in the network. In the model, the excitatory signals increase the potential of the neuron by one while the inhibitory signal decreases it by one. The existence of a solution for the non-linear signal flow equations for NNs modeling excitatory signals is proved. This architecture successfully models images and their texture.

Chart 10: Features of Arch. Max

- NP do not influence # iterations to select the winner neuron
- Every computation in a neuron is local
- Computation not controlled by a central processor

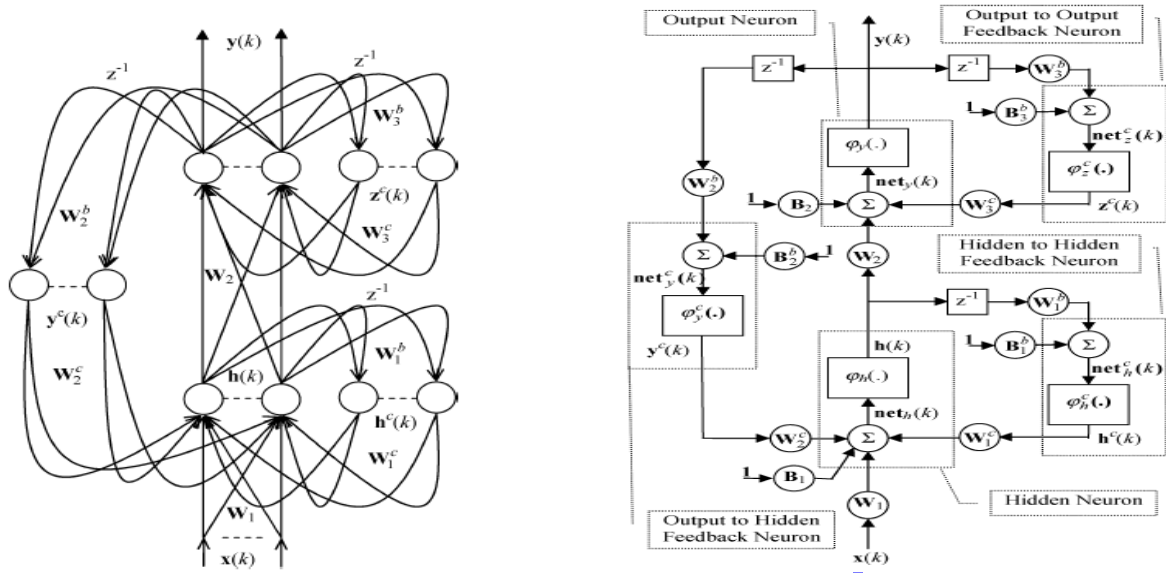
4.10 Block_feed back. NN: It exhibits flexibility in the specification of network architecture. It is a discrete time dynamic NN with feedback paths between layers.

Chart 11: Block F.B.NN

- It is a general model to describe different architectures.
- Each block consists of multilayer perception satisfying the trainability condition.
- Four different connection of the block result in a train of NNs, satisfying the trainability condition.
- Back propagation training results in universal approximation and Bayesian classification properties for block feedback NNs.
- These properties are extendable to dynamic system. Hopfield NN is applicable only to systems at equilibrium state. These nets are also called non-fix point NNs, different from fix point NNs.
- The NNs with delays are called here dynamic rather than recurrent;

to emphasize that output evolves in time as a result of time varying input

Architecture: The blocks are connected by different types of connections viz., cascade, sum, split and feedback (Chart 11). In each connection, one or more number of blocks is embedded. Each block consists of connection layer represented by weight matrix and functional array, consisting of transfer function etc. BP_NN is a specific instance of this family. It surmounts the limitations of fixed architecture. Within this frame work, a large number of different architectures are possible and can be trained. A multi-feed-back layer network is depicted in Fig.6.



Architecture **Layers**

Fig. 6: **Multifeedback-Layer Rec_NN (**courtesy of Ref : 254)

4.11 Pipelined_Rec_NN:

It is a complex modular NN consisting of a number of small scale Rec_NNs (Fig.7). It represents non-linear Wiener-Hammerstein cascaded systems.

Architecture. Pipelined _Rec _NN: A pipe-lined_Rec_NN consists of 'm-number of Rec_NNs as modules. Each module has 'N' neurons. The (NPx1) dimensional external input is delayed by 'delay' times steps before imputing to the module 'm'. All the modules operate with the same 'W' matrix. The overall output of the signal of Pipe_Rec_NN is Y_{out} (n) = $y_{1,1}$ (n).

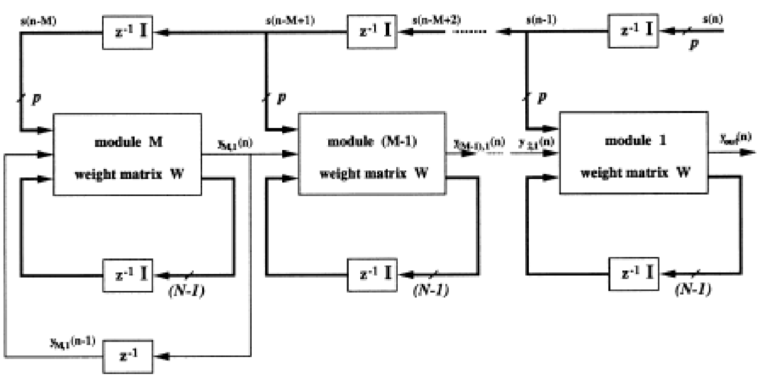


Fig. 7: Pipelined recurrent neural network. (Courtesy of Ref: 210)

Learning alg. Pipelined _Rec _NN: The learning algorithm for such large NN is very complex. The learning algorithms used are back propagation through time, recurrent back propagation and extended recursive L.S etc. The pipelined Rec-NN was trained with RTRL for joint processing adaptive non-linear equalizer for communication systems. It eliminates non-linear distortion in chaotic communication systems. It is better than Rec_NN.

4.12 Multi-feed-back-layer-NN: It consists of three feedback layers wherein both local and global recurrences are implemented. BPTT algorithm for both online and off-line training and Levenberg and Marquardt (LM) with trust region approach is used to optimize the Ws of NN. The prediction of chaotic time series and identification of non-linear dynamic system with the current Rec-NN is superior to the earlier models.

Architecture. Multi-feed-back-layer-NN: It has a SLP forward architecture. The recurrent connections are layer wise from output to output, output to hidden and hidden to hidden layers. The data flow in the feedback layer is as follows. The weighted output from a layer passes through an intermediate (context memory) layer with equal number of neurons. The weighted data passes through a transfer function (TF) and the weighted output from the memory layer is passed through the destination layer. If the destination layer is same as the layer from which the feedback is generated, it is referred as layer wise self-feedback connection. On the other hand, if the destination layer is any other layer (other than itself), then the feedback is called global layer wise. In a general feedback of a neuron, the output of the neuron is weighted and is passed as input to a memory neuron. This net signal after operated by a TF, the output of memory neuron is generated. It is weighted and connected to the input.

Datasets. Multi-feed-back-layer-NN: It is applied to identification of multi_input_multi_output (MIMO) system, non-linear plant and prediction of NARMA processes.

*Datasets.*Mackey-Glass.*Multi-feed-back-layer-NN* : It is superior to MLP (4-16-1) NN with tanh and linear TFs in hidden and output layers. The initializations of Ws were done by Nguyn-Widrow method and refinement by LM with trust region approach.

4.13 Liu-Wang-Rec-NN: It is proposed for a solution of a quadratic mathematical task. The solution is optimum for convex object functions with equality (linear) constraints. A quadratic function in two variables with linear equality and inequality constraints is also solved. The non-linear (sixth order polynomial and exponential) function with constraints also converges to the optimum with 3 iterations.

*Architecture.*Liu-Wang-Rec-NN*:* It is a one layer Rec-NN with discontinuous hard-limiting TF.

Dataset. Titanimum. Liu-Wang-Rec-NN: The titanium data was analysed earlier with SVR using Gaussian function as kernel. Liu-Wang-Rec-NN successfully trained the dataset.

4.14 Echo state network: The echo state network (Echo_State_NN) is a recurrent neural network with a sparsely connected random hidden layer. The weights of output neurons are the only part of the network that can change and be trained. Echo_State_NNs are good to (re)produce temporal patterns.

4.15 Rec-NNs with autonomous learning*:* It is a fully connected recurrent NN [231] and dynamic with continuous inputs. The number of neurons is equal to the sum of number of inputs and outputs. Each neuron can be an input/output/eventually both at the same time. All neurons take part in the calculation of I/O mapping. Sufficient conditions have been obtained to ensure the asymptotic stability. The comparison of RTRL with constant learning rate parameters and other similar ones clearly showed the efficacy of the current algorithm. They reach optimal values with respect to minimization of MSE. This NN can be considered as a low pass filter with adoptive cut frequency.

Datasets. Rec-NNs with autonomous learning: A second order system varying with time, non-linear discrete time series, MIMO system, evolution of chaotic intensity pulsations of far IR-NH_3 laser and the heat process datasets are tested with this NN.

4.16 Hybrid Rec-NNs

Reliable and possibly robust one-step and multi-step-ahead forecast is the need of the hour. The hierarchical hybrid algorithms meet the requirement with, of course, limitations of hardware, software expertise, benefit_to_cost_ratio analysis for a task. Binary and ternary hybrid rec-NNs (Table 3) are developed by combining with fuzzy information system fuzzy information system (FIS), self-organizing map (SOM), wavelet etc. which have unique advantages.

Table 3: Binary and ternary hybrid Fuzzy Rec.NN [254] ~~Rec-NN/44~~

Hybrid NNs	System ($$$_NN)	Function	Paradigm 1	Paradigm 2	Paradigm 3
Binary	Rec-neuroFuzzy	Long-term non-lin processes	FIS	Rec_NN	
	Rec-Dynamic_ Fuzzy Rule		Dynamic fuzzy rules	Rec_NN	
	Rec-Dyn-FLS	Non-singleton generalizations	Dynamic fuzzy logic system	Rec_NN	
	Rec-TSK		Takagi–Sugeno –Kang (TSK)	Rec_NN	
	Rec-FIS	Additive delay & BP	FIS	Rec_NN	
Ternary	Rec-SO-FI		SOM	FIS	Rec_NN
	Rec-TSK-wavelet		TSK	Wavelet	Rec_NN

M3-forecast-competition contains 756 data sets in demographic, finance, industry, macro- and micro-fields.

▶ NN with self feedback + Generalized_Rec_NN

Amiri [326] proposed a hybrid NN with self-feedback and Generalized_Rec_NN as components to model recall of associative digital patterns. The analog, digital patterns and grey scale pictorials are stored in stable fixed (SF) points of SF-NN. Generalized Rec_NN is used during retrieving process. This model is robust to high noise and does not have spurious attractors. It is better than recurrent associative memory models.

▶ Rec-FAS-ARTMAP

Architecture: A recurrent structure is proposed in Fas_ARTMAP. Feedback (recurrent) connection of output module to input through a context filter module is a new addition. The affective input of ART-x module is input.Artx = [B, C]. B includes features of the pattern, C is contextual (relationship among patterns in sequence) information. Partial- or full- information is used. It is used for pattern recognition tasks with sequence importance (KB.3).

KB. 3:	**Order of presentation of patterns**
If	No sequence dependency or contextual information is absent
Then	Patterns can be presented in any order
If	There is sequence dependency
Then	Different orders of presentations produces different results
	Remedy : Recurrent connections

▶ Rec-Fuzzy-NN

Rec-Fuzzy-NN is used for a long term prediction of non-linear processes like neutralization of acids. The knowledge of the process and input/output data are made use in dividing process/operation into several fuzzy regions and also to initialize W matrix of the fuzzy layer. With the input/output data and membership functions (mfs) of fuzzy operating regions, local linear models are trained. The object function is the minimization of the long term prediction errors. The local linear models are then passed through the defuzzification step using center of gravity (COG) procedure resulting in a global model.

Architec.Rec-Fuzzy-NN: It consists of five layers. The process variables are the input layer. In fact, this layer acts as a transmitting pipeline to the fuzzy rule development system (layers 2 to 4).

Layer 1 – Input: In the input layer, the accumulation operator, transfer function and the synaptic strength (w) of each neuron is unity. Thus, whatever is inputted is truly transmitted without any change i.e. copying operation.

Layer 2 – Fuzzification: In the fuzzification layer, the number of neurons for each input variable is equal to the number of fuzzy regions contemplated into which the crisp (floating point) domain (range) is to be divided. Thus each neuron corresponds to particular fuzzy set. The output gives the membership function here, sigmoid, its compliment and radial basis function.

Layer 3- If-part: The inputs to the third layer are the fuzzy sets which correspond to the combination of operating regions of different variables in the process. The output is a combination of these fuzzy regions. Fuzzy intersection is used along with the available knowledge.

Layer 4 (Consequent/Then part): A neuron in the fourth layer represents particular operating region. The weights in this layer are the local model parameters. The consequent part(s) is tested here using the inference algorithms.

Layer 5 (Defuzzification): The input to defuzzification layer is in the fuzzy form of the action/consequent. It is converted into the crisp form by the defuzzification procedures like COG. The number of IFs or the neurons in the rule layer corresponds to the number of fuzzy rules generated in the fuzzy-NN. In fact, the number of neurons in the consequent layer should also be equal to those in the rule layer. But, some rules may have the same consequent and hence the number of neurons in the THEN-layer is less. But, one can count the number of connections to all the neurons in the consequent layer which can of course equals to the number of rules.

▶ SOM + Rec-NN

A temporal-Kohonen-map-NN is generated by recurrent connections implemented in static Kohonen-SOM. Now, the response not only depends on current output but also on the activations of neural map in the preceding stage.

▶ Prob-NN + Elman

Ganchev et al. [247] hybridized probabilistic and Elman NNs (Chart 12). First, it is trained like Prob-NN, followed by differential evolution in optimization of W(eights). The text independent speaker identification is modeled with this hybrid Rec_NN.

Chart 12: Architecture. Prob-NN + Elman:

pro-NN	IL → pattern L→summation (or classification) L	→	output L
pro-NN + Rec Layer	IL → pattern L→summation (or classification) L → Rec L	→	output L

▶ RBF+ Elman

Hsu [28] proposed an Elman-based self-organizing RBF_NN. A chaotic TS and an inverted pendulum data are analyzed.

Arch. Elman + RBF: The initial architecture is without hidden neurons. Mahalanobis distance based online learning involves structure and parameter up gradation.

▶ NARMAX + Elman

Ardalani-Farsa et al. [81] trained Elman RecNN with embedded phase space points. The residual of predicted time series exhibits chaotic trend. Now, these residuals of TS are reconstructed according to embedding theorem. Another Elman is trained to predict time ahead values of the residual time series. This residual analysis is repeated in iterative mode. At this stage, NARX_NN is trained and used to model the relationship between predicted value and residuals and original time series. This method is applied to Mackey–Glass, Lorenz equations and Sunspot time series (Fig. 8) to evaluate the validity of the proposed technique. The hybrid Elman emulating NARMAX is successful in predicting chaotic time series more accurately when compared to other prediction approaches.

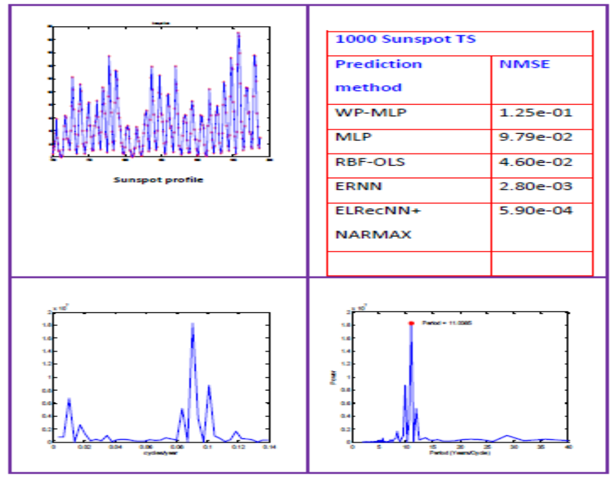


Fig. 8: Sun spot data modeling with hybrid Elman NN

▶ Discrete_PSO_Wang + Elman

Architecture: The structure with low architectural complexity is achieved with PSO. The parameters of each structure are minimized with IPSO, which employs a new velocity upgrade procedures [43]. A gene evidence for mutation strategy is to diversify the swarm and improve convergence.

Data: The results of simulated Mackey–Glass/ CATS time series data and real-life task of thermal system in a 600-MW power plant show better prediction accuracy, generalization and yet have less messy architectures.

The applications of hybrid Elman_NN in diverse disciplines are described in Table 4.

Table 4: Applications of binary hybrid Elman_RecNN

Model	Function	Task	Ref
Elman + Wavelet	balanced and unbalanced short circuit faults transmission line fault location model which	for selecting distinctive features about the faulty signals to determine the fault location occurred on transmission line rapidly and correctly	99
Elman NN RBF	real-time pattern classification Restoring the memory of past events.	Intrusion detection system secure model anomaly detection and misuse detection Data: intrusion detection sponsored by U.S. Defense Advanced Research Projects Agency (DARPA)	94
Discrete wavelet transform + Elman	Doppler signals were decomposed into time–frequency representations	Diagnosis: ophthalmic arterial (OA) and internal carotid arterial (ICA) diseases with Doppler signals	126
Wavelet multi-resolution	Load series are decomposed to different sub-series, showing different frequency characteristics of the load	Short-term electrical load prediction 1-day-ahead	47
Elman	Optimally designed Tr: static BP		
Two small Elman-NNs	Intra-cellular variables	Fed-batch recombinant fermentation	143
Auto associative-NN	Filter the noise		
FF-NN	Feed forward network as the controller		

Wavelet Elman	Local time–frequency transformation Noise removal high quality Non-lin multivariate calibration prediction	Overlapping voltammogram	128
Wavelet Elman	To process the measurement data by extracting the approximation coefficients of sensor measurement data. To identify sensor faults	Fault detection and diagnosis in of air handing unit	84
Discrete_PSO Elman	Structure of a RecNN Parameter learning	Simultaneous structure and parameter learning	43
Wavelet transform Elman	adaptive forward linear prediction;	Denoising in fiber optic gyroscope	44
PSO Modified Elman_NN	evolution of network structure, weights, initial inputs of the context units feedback coefficient	control for Ultrasonic Motors (USM)	116 29
NARX+ Elman MOGA—Pareto-opt	functionality & quality in wiping blade		
Elman + wavelet decomposition	monitoring environmental variables	treatment of underlying temporal structure at low frequencies	106

Quantum-NN + Elman NN

Li et al [32] proposed hybrid Elman-RecNN with quantum neurons (Chart 13, Appendix A2.a). The principles of quantum physics are used to account for interactions of qubit and classical neurons. The context (i.e. feedback from hidden) layer weight matrix is extended into hidden layer weight matrix. During up gradation these weights, more information of input profiles are used. The (sub) optimum architecture is arrived at by GA. This hybrid system has high accuracy.

Chart 13:[Quantum + Elman]NN

Layer	Neurons
Input	Quantized input
Quantum map	m1 qubit
hidden	m2 classic
output	m-D real

Dataset.electricity_load. Quantum-NN + Elman NN: With quantized input of hourly historical load, hourly predicted target temperature and time index, highly accurate forecasting of short-term electricity load became possible.

4.17 Second order Rec_NN

Architecture: A two inputs and one linear output system [222] with two hidden recurrent neurons is developed. Here, the chromosome has 10 genes (weights and inputs).

Fitness Fn: The reciprocal of the average error in the training set is the fitness function and the best individual in the population has minimum error. The stopping criteria are the completion of maximum number of generations or all the examples are recognized. Stochastic sampling with replacement is employed to create intermediate population. The results with 2, 3 and 4 recurrent neurons tabulated (Table 5) show the efficiency of the method over RTRL.

Table 5: comparison of GA and RTRL in training

Lrn alg.	**# neurons**		
	2	**3**	**4**
RTRL	0.0000179	0.00002094	0.0000202
GA	0.0000076	0.00000123	0.000001072

4.18 Higher order Rec-neuro fuzzy NN

Theocharis [238] proposed a high-order Rec-Fuzzy-NN, an online algorithm. Mamdani's method is used for fuzzy-inference. The current value of the internal variables is obtained from a number of past values of the fixed rules. Hence, multi-step-ahead-prediction of the internal variables is obtained at the consequent part. It is based on recurrent SOM fuzzy inference system NN (Rec-SOM-Fuzzy-NN). An internal feedback loop is introduced by circulating the firing strengths of the rules. The underlying topology is a hyper-graph. In other words, it allows weighted hyperedges connecting more than two neurons. The number of neurons it connects represents the degree of hyperedge. The largest hyper edge degree in the topology is called the order of a High-Order-Rec_NN.

Architec.High-order-Rec-Fuzzy-NN : There is a feed-back loop to the antecedent (IF part) layer. It allows firing of rules to be dynamically determined based on their past values. The feedback path consists of both context nodes and associated feedback nodes. The unique advantage of this NN is that it has both spatial and temporal processing capabilities. The feed-back connections from context layer nodes memorize the history of firing the rules. FIR-synaptic filters are introduced in the context nodes. The advantage is enhanced by temporal processing capabilities due to high order feedback loop leading to a higher order NN with increased temporal ability. The dynamic rules are decomposed into external and internal rules. The external rules represent the output of the model. Internal rules refer to the evolution of internal variables and inter connections of the NN at different time periods. A set of dynamic fuzzy-rules for the consequent part of the rule adds a new capacity of multiple-steps-ahead predictions of internal variables. This is a new feature not available in many fuzzy information systems (FISs) and fuzzy-NNs.

Datasets.speech recognition.High-order-Rec-Fuzzy-NN: It is applied to speech recognition task with different types of noise structures. Due to adaptive noise cancellation, higher levels of speech enhancement are achieved. The modeling of 1500 training samples for the utterance of digits '1' to '6' by an English male with this hybrid NN exhibited superior performance due to multi-stage inter connections and the enhanced internal dynamics.

Datasets.industrial process.High-order-Rec-Fuzzy-NN: The output of an industrial plant is a function of three preceding outputs and two past inputs. Earlier the system was studied with Rec-NN, memory-NN, Rec-SOM-fuzzy-NN. Theocharis [238] reported High-order-Rec-Fuzzy-NN with 2 fuzzy rules excels earlier reports and IIR-MLP and DC-NN. The system identification represented by the difference equation is superior with the High-order-Rec-Fuzzy-NN compared with IIR-MLP or Dynamic-NN.

4.19 Hierarchical Rec-NN

Here Rec-NNs are sparsely connected together through bottlenecks with the idea to isolate different hierarchical functions to different parts of the composite network.

4.20 Recursive_NN for extraction of rules from trained NNs

The rule extraction procedure depends on dividing the continuous state space of Rec-NNs with sigmoidal discriminant function into discrete partition. Setiono [270] recently proposed a recursive NN to extract rules form a trained MLP-NN for a classification task. The values of inputs (antecedents) are discrete or continuous numerical values. In the first step, the trained NN is pruned to remove irrelevant /redundant neurons and connections between layers. The decision tree from C4.5 algorithm is used to generate rules with discrete antecedents. The final rule set from the current Rec-NN algorithm (Alg. 5) is hierarchical. The rules at the deepest level only have antecedents consisting of linear combinations of continuous variables. It enhances the understandability of the rule compared to simple If-Then-else structure. It is believed that this approach renders black-box-NNs into a white-box one through greyer stage.

Alg 5: Algorithm of RecursiveRuleExt R

```
[Rules] = RecursiveRuleExt R (Dataset,D1,C1)

Dataset1: Data set with NP patterns and (D+C)          features Rules : If-Then rules
C1  : Continuous features                               D 1: Discrete features
Step      -2     Train Dataset with NN
Step      -1     Prune the features in trained NN
           0     D, C: are the pruned features. S : Data samples correctly classified by the pruned NN
Recursive Rule extraction of a trained NN
Step  :   1      If       D = null
                 Then     Generate hyper plane to split the samples (S) according to values of  continuous features (C )
                          Stop
                 Else     Use only the discrete features (D)
                          Generate set of classification rules
                 For      Each rule Ri generated
                          If      Support(Ri) >  δ1  & error(Ri) >  δ2
Step: 2                   Then    SRi : Set of data samples satisfying Rule (Ri)
                                  DRi : Set discrete features which do not appear in the IF part
                                        of the rule (Ri)
                                  If      DRi = 0
                                  Then    Generate a hyper plane to split the samples based on values of continuous
                                          features
                                          Stop
                                  else    [Rules] = RecursiveRuleExt(Dataset,D1,C1)
                 End for
```

Dataset.Credit card: The German credit card (Card3) dataset containing 51 variables is trained with SLP-NN. After pruning the network architecture is 7-1-2 and the rules are generated with Re-RX. Card 2 dataset was trained with Prob.NN and the pruned NN has the architecture 6-2-2 (Table 6).

Dataset. credit holder: The datasets, Bene-1 and Bene-2 are from major financial institutions in Belgium, the Netherlands and Luxembourg. The credibility of customers in repaying loans was judged based on the pending installments for more than ninety days. The data was analysed and the rules are extracted.

5. Emulation of standard mathematical techniques by Rec.NNs

Emulation of earlier standard techniques by a latest paradigm is a testimony of its imbibing character (Fig. 9) and becomes sought after for its additional features.

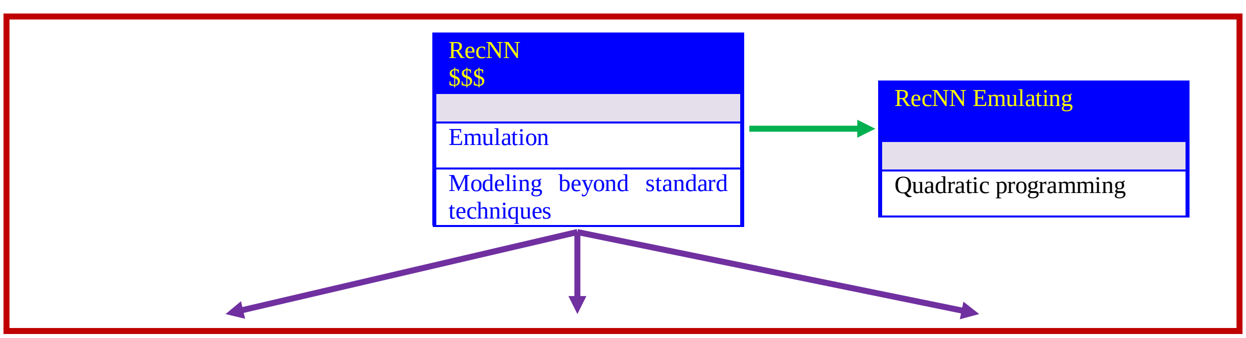
Table 6: Credit card data analysis

(a)Accuracy of NNs

Data set	NN	% Accuracy	
		Tr	Te
Card3	SLP	87.26	88.95
Card2	PNN	89.38	86.05

(b) Hierarchical Rules generated by Re-Rx for Bene data

```
If    purpose =  cash loan &
      marital status = not married  &
      known-client = no
      If       Owner-of-real-estate = yes
               If       Term-of-loan < 27 months
               Then     Customer = good-payer
               Else     Customer = Defaulter
      Else     Customer = Defaulter
end
```



RecNN Emulating $$$_TimeSeries	RecNN Emulating $$$_models	RecNN Emulating $$$_Filters
NARMA	William Zipser	Adaptive-IIR
NARMAX	Hammerstein	FIR

Fig. 9: RecNN emulation of standard mathematical/statistical procedures

5.1 Universal function approximators: Recurrent multilayer perception [219] can also be considered as universal function approximator, although explicit proof is not available.

5.2 Optimization task*:* Rec-NNs are used in solving optimization task. A primal-dual-NN based on linear variational inequalities to solve LP/QP tasks on line is developed. In order to realize the algorithm on application-specific-integrated-circuit , MATLAB Simulink modeling module is used. The results substantiate theoretical results.

5.3 Inverted Pendulum: The pole is kept up in the conventional solution of an inverted pendulum [202]. Rec_NN is used in solving the swinging of the pole of the inverted pendulum and stabilizing the pole, locally at the upper equilibrium point. The transition between equilibrium points can be realized by static and dynamic neural controllers parameterized by FF_NN or Rec_NNs. This approach is unique in considering output feedback and the controller is static in FF_NN while dynamic in Rec_NN.

5.4 State estimation:

Hung [351] investigated the applicability of Rec-NN with time varying delay in state estimation. A delay partition approach is proposed and gain matrix estimator is obtained by solving linear matrix inequality. This model is better compared to earlier procedures.

5.5 Filters

- **Infinite input Response (IIR) filter**

It is another type of dynamics where in the input at a point of time continues to influence the response of the system add infinite in time. The only way to remove the barrier of influence is resetting.

IIR-Rec-NN: In IIR-Rec-NN (Fig. 10), the Ws are refined with infinite impulse response (IIR) filter [238]. The architecture is 2-7-1. Here WIH and WHO are optimized with MA and AR of order [1, 1] and [1,2]. The diagonal-Rec-NN is a Rec-NN with the hidden layer containing self-recurrent neurons**.** The dynamic-Rec-NN model used is 2-10-1.

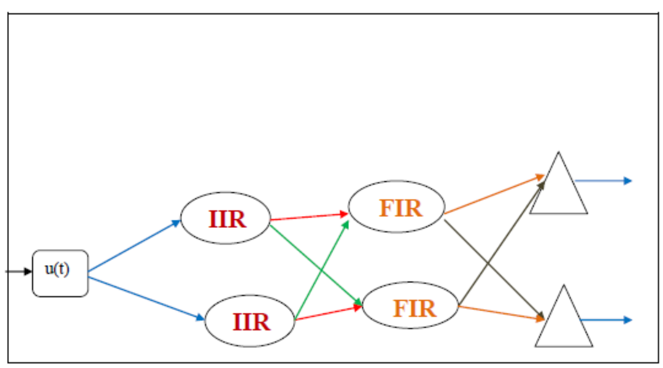


Fig. 10: FF-NN with IIR and FIR neurons

Cascade-Rec-NN [280]

The cascade structure of the network has non-linear neurons with IIR filter in the first HL and linear neurons with FIR filter in the second HL. The neurons at the output layer receive excitations from both neurons of the previous layer and external input layer

5.6 Emulation of time series models with Rec-NNs

The state_of_the_knowledge (SOK-) _of_the- time_series_models (TSM) is briefed in appendix A4. Rec-NNs not only mimic the popular MA, AR, ARMA, NARMA, NARMAX, Weiner , Hammerstein and Volterra time series models, but more complicated profiles can also be modeled for the m-step-ahead prediction. In general, they function as polynomial/non-linear adaptive filters. In fact, there is no transfer function of a non-linear filter in the frequency domain. That is the reason why in the design of a non-linear filter, it is the transformed into a constrained optimization in Fock space.

Two popular ways of inclusion of recurrence in FF-NN are inclusion of feedback from the output of hidden layer (HL) and/or output of output layer (OL) in addition to self-feed-back. For temporally (i.e. with respect to time) or spatially varying data, delays of any order (1 to ndelays) will result in dynamic NNs.

The simulated and real time series data sets analyzed in literature is very large and a bird's eye view even in a sub-discipline is a herculean task. The simplest example to cite for an amateur is inverted pendulum [202] and the other is XOR gate for a two-class problem [233]. The intense research with Rec-NNs showed their supremacy in QSAR with molecular descriptors, non-linear filtering, modeling stochastic processes, solution of second order differential equations, constrained quadratic equation and storage/retrieval of corrupted/similar signals [225], system identification of a high order linear/nonlinear systems and segmentation of DNA [52]. Further, Stock price index, GNP, market index, German DAX 30 index, exchange rate, distribution and consumption of electric power, weather forecasting, ozone/nitrate now-cast, IR-laser and EEG are noteworthy applications. Bench mark items include sunspot data, gas-flow, Lorenz attractor, and Mackey and Glass data generated through differential equation, chaotic series by delay DE, second order DE in m-dimensions. The time of observation scale is in hours, days, trimesters and years. The details of a few typical case studies in different disciplines follow.

- NARMA

NARMAX-Rec-neurons in a standard NN emulated NARMA analysis. The input and output to the system are delayed response. When NARX-NN is unfolded in time, the output delays will appear as jump-ahead connections in the unfolded NN. Interestingly, these jump-ahead connections provide a shorter path to propagate the information of gradient. It is less sensitive to long term effects as sensitivity of NN reduces for long term dependencies. Weiner system can also be represented by NARMA model.

Datasets.NARMA.Time series.Multi-feed-back-layer-NN: A non-linear ARMA process represented by the difference equation is modeled by dynamic non-linear non-singleton FLS, rec-fuzzy-NN and Multi-feed-back-layer-NN. With noise of varying magnitude (0.3 to 0.7), Multi-feed-back-layer-NN has lower MSE, SD and training is completed with very small number of epochs.

- Non-linear Dynamic process

Any stochastic process generated by a finite order non-linear model can be optimally estimated by a suitable back-feed NN. From the universal approximation property of MLP-NNs, F and mu can be approximated by a three layer FF-NN. It is also possible to build a network approximating Q**(X) = HX**. If the time dependency is also considered, addition of feedback connections results in a Rec_NN accepting the input values (u(t)) and computing that at (t+1).

Chart 14: Performance of Pipelined-second-order-Volterra-Rec-NN

- Parallel execution in a pipe lined fashion
  - Improvement in computational efficiency
- Recursive second order Volterra expansion

Volterra model

Yang [362] implemented Volterra second order model of dynamic (FIR, IIR) system in SLP NN. The cross product terms are added to enlarge the non-linear binary interactions between the variables. This NN employed PSO-algorithm in training Ws. The superior performance of this hybrid model is evident in discrete bilinear and non-linear time varying dynamic simulated datasets. The effect of Gaussian error is also investigated. NARMAX neuron with input signals viz. [u(k-1), u(k-2)] results the general form of Volterra series.

- Introduces Non-linearity
  - Increased efficiency
- Performance.
  Pipelined-second-order-Volterra-Rec-NN >> Pipelined-Rec-NN or Rec-NN
- Predictions
  - Non-linear colored signals
  - Non-linear channel equalization

Pipelined-second-order-Volterra-Rec-NN

Zhao [369] proposed pipelined-second-order-Volterra-Rec-NN.

Architecture.: It has a number of simple small scale second order Volterra Rec-NNs.

Training Alg.: A modified RTRL and a heuristically enhanced gradient approximation algorithm is used to train this NN. The performance is briefed in chart 14.

Wiener Model

Rec-NN emulates Wiener model. When the noise is not correlated with the input signals, an infinite length of the exact representation of the equation is necessary. Assuming the mild condition that a finite-degree-of-polynomial-steady-state-characteristic is inadequate, the output of the non-linear model can be split into a set of sub-models. Wiener model consisting of a linear dynamic system followed by a zero-memory-non-linearity represents the time series data.

Hammerstein model

It is a parametric model wherein a linear dynamic system succeeded by a zero memory non-linearity. The noise can be added to output of a SISO Hammerstein system. Rec-NN emulates Hammerstein model. Both Weiner and Hammerstein models are combined resulting in complicated Block-stochastic model used to compensate each other. Hybrid NNs resemble Weiner and Hammerstein stochastic models.

William Zipser model

Rec-NN emulating William Zipser model consists of a hidden layer and output layer. The input layer has both feed-back signals and experimental time-delay observations.

Time_series with seasonal patterns

Gheyas and Smith [68] reported a method wherein an ensemble of generalized regression NNs is combined in a Generalized_Rec_NN. The forecasting of TS with seasonal patterns is excellent with ensemble Gen_Rec_NN compared to 11 algorithms including RecNN models on real data sets.

5.7 Emulation of Mathematical programming tasks

Rec-NNs were also proposed for solution of convex quadratic program, linear piecewise equation, non-linear convex-problems with linear constraints, linear projection equations etc.

Quadratic Programming

Quadratic programming methods suit to model image and signal processing, regression analysis, parameter estimation, robot control and filter design. Signal processing for adoptive beam form and control of robotic motion are real life tasks. A primal Rec-NN with two layer structures was proposed. The solution is approximate, since it has a finite penalty parameter. Xia [227] surmounted the problem by using a primal dual Rec-NN. Xia et al. [227] applied a dual NN for kinematically redundant manipulator. Earlier, the quadratic programming problem was transformed into piecewise equations and solved each with Rec-NN. The architecture consists of a single layer with a low complexity. Liu and Wang [278] proposed RecNN with one-layer using a discontinuous hard-limiting activation function for quadratic programming. The number of neurons in the neural network is equal to the number of variables in the optimization task. The NN is guaranteed to optimal solution of any type of quadratic programming tasks. But, the objective function has to be strictly convex on a set defined by the equality constraints.

Linear projection equation

Many constrained optimization problems can be translated into equivalent linear projection equations. Tank and Hopfiled were the first to propose a NN to solve a linear programming problem. They mapped the problem into a closed loop circuit. Of late, it inspired many researchers. Xia and Wang [379] proposed a Rec-NN to solve linear projection equations. This NN is globally convergent to the solution and is exponentially convergent if the matrix is positive definite. The Rec-NN has two layers. This solved convex quadratic task, quadratic optimization problem with bound constraints and linear constrained jobs. The applications of linear projection equations in real life tasks include modeling of traffic network, competitive processes, piece-wise-linear-resistive-circuits and vibrational inequalities. The pivoting and iterative methods (interior-point, projection-gradient) are popular numerical procedures. The hardware implementation requires summation units, integrators and waited connectors, but not analogue multipliers for variables or the penalty parameters.

6 Applications RecNNs

AI-2 methods are now an integral part of system deriving information from data as a sole procedure, preprocessing, post-processing and/or one of the modules of core data crunching. Statistical procedures, models from first principles, a priori trends/model/constraints, heuristics, E-man [5-7,9] are all in any combination march to arrive at a set of sub-goals, intermediate checks and the target prime goal. The inspiration for Rec_NNs is from neuro-/population-biology and evolutionary theory with many assumptions and approximations. During the last quarter century, recurrent NNs proposed for dynamic processes/ associative memories and variation of response/parameter in spacio-temporal regime spread its wings to diverse fields like system identification, signal processing, forecasting, time series analysis, non-linear dynamics in computer science, engineering, process chemistry/technology and stock market/forex. The increased complications gave way for simultaneous refinement of training procedures and newer architectures. The rec-NNs with varying time delays model many processes in neuro biology, population biology and evolutionary theory, of course, with a set of relevant observations/constraints. The time delay corresponds to the finite speed of the axonal signal transmission. If the time delay is zero for the entire time series, the model reduces to the popular statistical procedure.

6.1 Bibliometrics

Papavlasopoulos et al. [79] used Elman-RecNN to extract scientific impact factor of journal combing many existing ones from a database of impact factors of cell biology journals.

6.2 Nuclear power plant

Ayaz [148] modeled nuclear power plant data at Borssele under wide range operational conditions with Elman, Jordan, MLP_BP NNs. With the learned NN, the reactor operation is followed by NNs. Şeker et al. [149] used Elman Rec-NN in condition monitoring in rotating machinery detecting anomalies high-temperature gas cooled reactor and motor bearing damage from coherence function approach.

6.3 Satellite-Altitude sensor: Chen and Shen [44] used Elman-RecNN for de-noising i.e., eliminating effect of variation of temperature of environment in Fiber optic gyroscope (FOG) with light weight and high reliability. FOG is critical as an attitude sensor in satellite and automobile.

6.4 Environmental Sciences

The ecosystem dynamics is a complicated web of multitude of chemical/physical processes in multiple phases under a variety of surroundings. Development of mechanistic models from first principles is formidable and hence, resort to black box paradigm gained importance [120].

.

Upper atmospheric: Martin et al. [131] forecasted electron concentration distributions in the 150–600 km altitude range above Arecibo, Puerto Rico. The incoherent scatter radar data and geomagnetic index are the input to NNs viz. Elman_RecNN and SLP. The basic data are from Arecibo Observatory and National Space Science Data Center covering two solar cycles. This paved way to forecast only upper atmospheric parameter distributions taking into consideration of daily, seasonal, and solar cycle variations.

Geomagnetic storm index: Watanabe et al. [151] developed Elman-RecNN to forecast two-hours-ahead geomagnetic storm index, which is in operation since April 1998. The input is velocity and density of the solar wind, the magnitude of the interplanetary magnetic field (IMF), and x, y, and z components of the IMF. Lundstedt [166] reported the results of prediction of solar wind parameters and geomagnetic indices.

Magnetic levitation control: Chen et al. [98] developed a combined two Mod-Elman-NNs to control highly non-linear and unstable MIMO magnetic levitation module. A discrete-type Lyapunov function is used for convergence analysis.

Weather forecast: Maqsood and Abraham [123] introduced an accurate weather forecast model for Vancouver, Canada. With a yearlong daily temperature and wind speed data, Elman-RecNN and MLP were trained with Levenberg-Marquardt algorithm and one-step-secant optimization methods. In the next phase an ensemble of NNs were generated with different data sets trained with MLP, RBF and Elman NNs. The arithmetic mean and weighted average of ensemble output produce an acceptable value. Maqsood et al. [130] compared feed forward RBF/ MLP, recurrent Elman/ Hopfiled NNs for 24hour weather forecast in southern Saskatchewan, Canada. Mellit et al. (112) used artificial intelligence in control of photovoltaic systems and to forecast meteorological data. A rough measure of the ionospheric energy losses or overall horizontal current strength is denoted by AE. Pallocchia et al. [114] forecasted AE from 5 min to 1 h from solar wind input at different time scales using modified Elman RecNN. Sfetsos and Coonick et al. [155] forecasted mean global solar radiation received by a horizontal surface on hourly basis with FF-RBF-, Elman_Rec-NNs and FIS. NNs are found better than clearness index approach.

Abbre-viations		**Full form**
HE	:	Homogeneous ensemble
HEW	:	HE with weighted averaging
HESimpA	:	HE with Simple averaging
HEStA	:	HE with static Weighted averaging
JE	:	Jacobs'Ensemble;

Arch.Elman_ Pallocchia: The hidden layer has four neurons and additionally four context (recurring from output of hidden layer) neurons. The image is input to this RecNN based on L1 solar wind IMF and experimentally measured plasma.

6.5 Pollution monitoring

The outdoor pollution speaks of air quality, an index of health of inhabitants. Elman-RecNN forecasted one-, two- and three-hour ahead SO_2 levels. Melilli, a highly polluted town in Italy [107], was subjected to evacuation operation. Brunelli [119] reported a pollution monitoring and management tool applicable for eight monitoring stations in urban area of Palermo (Italy). Here, Elman-RecNN modeled the time series data consisting of daily maximum values of SO_2, O_3, PM10, NO_2, CO concentrations during the period 1 January 2003 to 31 December 2004.

PM10: Siwek and Osowski [49] studied daily forecast of average PM10 in Warsaw by wavelet and NN ensemble modeling techniques. The individual prediction results of SVM, MLP, RBF, ARX, Elman and wavelet are combined in ensemble and another NN outputs integrated final predicted PM10 concentration with better accuracy.

Prediction of ozone: Salazar-Ruiz et al. [101] reported Elman-RecNN for one-day-head-prediction of maximum tropospheric ozone concentrations in the Mexicali, Calexico, California (US) with an input of daily means as well as a mean for the first 6 h of the day. Further, persistence, parametric approaches, SVM, MLP, ridge regression also were used and the recommendations of EPA-US were adapted.

Heat island: The materials used in civil construction and other activities result in urban heat island which perturbs energy balance of buildings, smog, SO_x, CO_x, NO_x levels, SPM etc. Gobakis et al. [60] made an intensive study of TS data of ambient temperature and global solar radiation for predictive models in Athens (Greece) with Elman, MLP and cascade NNs.

Greenhouse control: Greenhouses in general and their internal climate are complicated systems and obviously classical statistical techniques or procedures from first principles fail. Fourati [25] reported greenhouse control using Elman and ART2 NNs. The greenhouse data is clustered using ATRL. Each cluster is the input to MLP and output is multiple neural controls. In this model, data is divided into several time periods and control measures are modeled with Elman. The clusters of ART2 are used now in supervised mode to select suitable control (Fig.11).

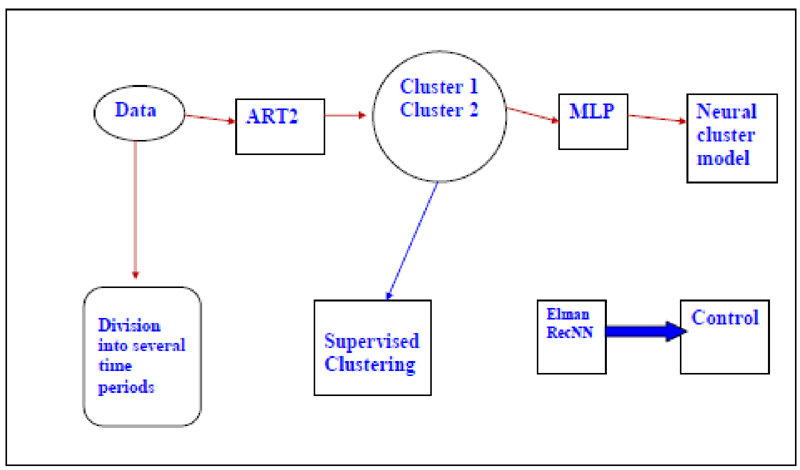


Fig. 11: System flow diagram of greenhouse control model with multiple paradigms (Courtesy of Ref 25)

Prediction rain flow: Thiery et al. [110] trained Rec_NN with LM to predict rain flow in Têt catchment area, the main river of the Pyrénées-Orientales department (Southern France).

Evaporation: Nourani and Fard [50] studied evaporation rate of Tabriz and Urmia cities on daily basis from hydro-meteorological data using MLP_BP, RBF and Elman NNs. The physical models taken in to consideration in this study are energy balance, aerodynamic, and Penman models. MLP_BP model is superior to RBF- and Elman-Rec- NNs.

Basin water quality: West and Dellana [72] reported JENN and GMNN models for cumulative multi-period forecast of basin water quality accurately. These NNs bring down the cumulative five-period forecast errors to 50% and superior to exponential smoothing, ARIMA, MLP and TDNN.

Water quality: Xu and Liu [45] compared MLP_BP, Elman and hybrid wavelet_Elman (Appendix A2.b) in predicting water quality of ponds in Duchang county, Jiangxi province, China (Table 7).

Table 7: Comparison of NNs for water quality

Time	Observed	APE %		
		MLP_BP	Elman	Wavelet
7:59	0.730	17.081	13.332	9.668
8:59	0.994	28.397	32.333	6.310
9:59	1.125	40.715	38.601	4.679
10:59	1.190	34.205	20.356	4.013

Inputs : first half hour of the DO, pH, Temp, air humidity, wind speed, and solar radiation
Software : Matlab7.13
APE: Absolute percentage error

NN	Archit	%MAE
MLP_BP	6-4-1	17.464
Elman	6-4-1	8.438
wavelet + Elman		3.822

Sludge plants: Sainz et al. [140] reported a superior performance of Rec_fuzzy_ART (Rec_Fas_ART) NN compared to simple Rec_NN for activated sludge plant data. This model uses contextual information from a real wastewater treatment plant.

Prediction of fish catch: Gutiérrez-Estrada et al. [117] studied 1-month-ahead-forecast of anchovy catches in the north area of Chile with Elman, ARMA and their combination. An input of six previous months catches, the hybrid ARIMA (2,0,0) and seasonal Computational NN explained 84 to 87% of variance.

6.6 Chemometrics

- Foodomics

The effects of genetic transformations on chemical composition of foods are studied with advances in in transcriptomics, proteomics, and metabolomics in combination with bioinformatics and chemometrics. These technologies pave way to characterize genetically-modified organisms at the transcriptome, proteome and metabolome levels.

- Calibration of Multi_component multi_channel_response data:

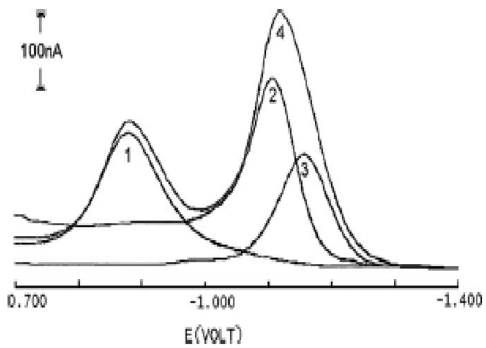
Gao and Ren [77] reported simultaneous estimation of nitrophenols (p-nitrophenol, o-nitrophenol, and 2,4-dinitrophenol) with highly overlapping spectral profiles in 350-450 nm region. LS–SVM learns a high-dimensional feature space even with less number of training samples. The prediction errors are lower for wavelet-Elman and LS-SVM compared to chemometric techniques PLS and MLR. Wavelet packet transform is a powerful de-noising method and Elman-RecNN (Table 8) has high quality calibration characteristics for overlapping spectra. This enables one to perform calibration and prediction even in water samples without a prior separation of chemical species.

Table 8: Calibration of substituted-phenols

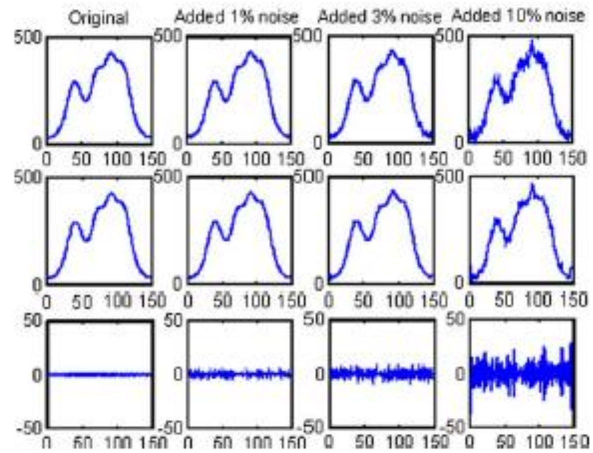
Chemometric Method	SEP $\mu g\, ml^{-1}$
LS–SVM	1.32
WPT–ERec_NN	2.39
ERec_NN	6.71
PLS	3.17
MLR	0.64×10^4

- Multivariate-multi response calibration

Ren and Gao [128] put forward wavelet based Elman RecNN and applied for the simultaneous estimation of Ni(II), Zn(II) and Co(II) by differential pulse voltammetry. This hybrid method showed better performance compared to other NNs and soft-regression procedures (Fig. 12). Wavelet packet representations of signals provided a local time–frequency description. Thus in the wavelet domain, the quality of the noise removal can be improved. The performances of the Wavelet Packet Transformation methods were compared with seven other filtering techniques in terms of root mean square deviations between reconstructed and original mean voltammogram. Elman recurrent network was applied for non-linear_ multivariate_multi_response_calibration to improve predictive ability. NNs are superior over factor-based methods.



(1): Ni(II); (2) Zn(II) ; (3) Co(II)



Methods	WPT	FFT	DCT	HFIR
RMS	0.012	0.016	0.021	0.066
Methods	BUTTER	BASSEL	CHEBY1	CHEBY2
RMS	0.021	0.977	0.025	0.016

Row	Profile
1	Experimental voltammogram
2	Model output
3	Residual

col	% noise
1	1
2	3
3	R10

Name	$$$_Transforms
Hankel	Hankel
Hadamard T	Fast Hartley
Hadamard	Hadamard
DCT	Discrete cosine
HFIR	FIR with hamming window
BUTTER	Butterworth IIR filter
CHEBY 2	Chebyshev type 2 IIR filter
CHEBY 1	Chebyshev Type 1 IIR filter
BASSEL	Bassel IIR filter

Chemometric method	(%) Relative SDP
Wavelet + Elman-RecNN	9.53
Elman-RecNN	9.82
PLS	12.3
PCR	17.0
TTFA	16.7
MLR	1.46×10^5

Fig. 12: Elman RecNN for electrometric estimation of metal ions (Courtesy from Ref 128)

Chemical industry

Bahar and Ozgen [82] predicted composition of a product in distillation column from temperature with modified Elman_Rec_NN.

Arch.Elman_ Bahar: It has two hidden layers with 20 and 34 neurons respectively. Hyperbolic tangent sigmoid transfer functions (tan-sigmoid) are used for the hidden layers and linear activation function for output layer (Chart 15). The feedback consists of designed estimator system.

Chart 15: Architecture of Elman_Bahar

Arch.Elman_ Bahar	#neurons	TF
HL1	20	$\tan sig(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$
HL2	34	
OL		$purelin(x) = x$

Fuel cells

Lobato et al. [89] used inverse NN to estimate initial conditions of characteristics of a fuel cell. The purpose of this study is to predict tortuosity value and voltage of cell versus density. Some of the factors influencing gas diffusion layer are mean pore size, air permeability, teflon content, porosity and hydrophobia level. Lobato et al. [95] modeled the influence of the conditioning/ operating temperature and current density on potential and cathode/ ohmic resistance in polybenzimidazole-polymer electrolyte membrane fuel cells in the working temperature range of 100 to 175^{o}C. MLP is found to be better than Jordan/ Elman RecNNs. Steiner et al. [67] proposed failure diagnosis and durability of polymer electrolyte fuel cells. It is related to flooding and drying out phases of water management. Proton Exchange Membrane Fuel Cell systems are now alternative viable current energy converters like internal combustion engines. They are used in stationary and transport applications in spite of insufficient reliability and durability. The limitations can be surmounted by intelligent corrective measures. The fault diagnosis in fuel cell is related to many parameters which is almost impossible to monitor with respect to geometry of stacks in fuel cell. The flooding diagnosis procedure was introduced based on the analysis of a residual (difference between experimental and estimated) pressure drop. The results under different experimental conditions including non-flooding and deliberately induced flooding situations endorse the success of the model. Solid Oxide Fuel Cell integrated into Micro Gas Turbine (MGT) is a non-linear multivariable strong coupling system. Wu et al. [62] proposed an architecture containing Elman as well as Jordan type recurrent connections and PSO is used for training Ws. The simulation results in MATLAB are successful.

6.7 Qualimetrics

A sub discipline of process chemistry in industrial sector includes time series of key production yields and process parameters to monitor the health of ongoing schedule and to raise alarming warnings in case of distractions.

Quality control of manufacturing process: Pacella and Semeraro [121] reported Elman-RecNN performs better than control charts for serially- (auto-) correlated TS data of manufacturing processes. This model does not require developing TS models separately.

Batch fermentation reactor: Patnaik [143] made a comprehensive study of β-galactosidase production by a recombinant Escherichia coli strain in a fed-batch fermentation reactor. The bioreactor is modeled as a hybrid system with two small Elman NNs for intra-cellar factors, an auto-associative NN to filter the noise and FF-NN for controlling. The process data covering a 12 h fermentation period was simulated by a 6-12-4 Elman NN with two extra- and two intracellular variables as outputs. Three types of inflow failure conditions were considered. The results are superior to MLP trained by BP.

6.8 Material science

Non-asbestos organic based friction materials: Xiao and Zhu [80] arrived at optimum non-asbestos organic based friction materials (Chart 16) by experimental design (Exp.Des), response surface methodology (RSM) and Elman-RecNN. Fractional factorial design (2^{k}) design (Fr.Fact.Des.) was used to study the effect of ingredients on friction characteristics. The dynamometer data at values of influencing factors arrived from Exp.Des was modeled with Elman-RecNN. The prediction of optimum friction was made and it led to optimum feature values. This approach is a trust worthy tool in manufacturing industry.

Chart 16: Optimum non-asbestos organic based friction materials

Friction characteristics	**Ingredients**
1st fading rate	Synthetic graphite
2nd fading rate	Potassium titanate
Speed sensitivity	Mineral fiber Calcium silicate Phenolic resin
(2^{k}) design of experiment	RSM with Elman RecNN

Rubber compounds: Karaagaç et al. [93] predicted optimum cure time of rubber compounds, the performance of methods being in the order NNs [(MLP), Elman (GRec_NN)] >> Equivalent cure concept.

Fermentation process: Niu et al. [61] proposed an ensemble of NNs to model the product concentration of fermentation of Nosiheptide. A separate Elman-NN learns Ws for each pair of training data subjected to bagging (resampling) procedure. The outputs of ensemble of these NNs were combined by weighted average method using PLSR. The end results show better accuracy and generalization ability compared to a single Elman NN for the entire training set.

6.9 Bio-signaling network

The biological signaling network is translated into ordinary differential equations (ODEs) and the solution is realized through a Rec_NN. An input neuron in NN corresponds to a unique biological signaling molecule. The trained weights are rate constants of biochemical processes. This NN behaves like functional block, or in order words, structural motif permits analysis of temporal behavior of the biological network.

-omics	[Gen_omics metabol_omics metabon_omics prote_omics transcript_omics]

6.10 Medical Sciences - State-of-art-of-health_care

The biochemical reactions, non-bonded interactions, H-bonding, charge transmission, cumulation / triggering of impulses are known [388] to a large extent. But, they cannot be modeled from first principles to visualize even at organ level leave alone in the single molecule in a single cell context. System biology, a conglomeration of biology_principles, biometrics, chemo-informatics and nature inspired approaches of white to black through grey box models is a viable probe. The compartmental bio-chemical models of system biology consists of hierarchical optimally organized (with generation, consumption) of ions, molecules, proteins, genes, chromosomes, hormones, cells etc. undergoing millions of chemical /physical (energy) transactions/transformations. The progressive march towards probing/understanding/visualizing/ emulating/ repairing odds of the processes in nature in twenty first century is at a cross roads of matured state-of-disciplines. Further, fine-tuning of micro energy level and ignored/ unveiled patches of past to complete the picture is awaited for future breakthroughs. Neural networks in general, Rec.NNs in particular along with experimental/ theoretical sparkling results made a mark in modeling, predicting, enhancing desired life characteristics and reducing /minimizing undesired wayward outcomes like HIV/cancer/ genetic defect and mental disorders.

Cosmotics- Skin care: Elman NN finds use in probing into women skin care with attributes viz. tone, spots, and hydration based on responses to questionnaires [66].

❄ Diabetes Mellitus

RecNNs are employed in tracing nonlinear trend of blood glucose metabolism in Type 1 Diabetes Mellitus (T1_DM) patients and insulin dynamics.

❄ HIV

HIV infection is a retrovirus kidnapping CD4+T, a fundamental part of immune system and has been intensively studied over last three decades [430-432]. The world health organization (WHO) that around noted 3 millions of people were with HIV all over globe at the end of year 2007 and 2 million died of AIDS in that year itself. The surrogate markers used in HIV treatment are CD4+ T-cell counts and plasma HIV-RNA levels (i.e. viral load). The number of infected cells is an unknown parameter in clinical practice. The measurements through bio-medical and clinical data are expensive, time consuming and complicated. NN models are reported for HIV infection monitoring. A black box model with concentration of infected (T*)/ non-infected (T) cells as well as the viral load in the blood count was proposed. The recurrent higher order NN models trained with EKF are used to simulate state estimation tasks in HIV. The results of these state estimation by NN models provide ways and means to probe into immune markers of high relevance to clinicians.

❄ Cardiology

Ubeyli [97] classified ECG signals into normal/congestive heart failure/ ventricular tachyarrhythmia/ atrial fibrillation beats using Elman-RecNN trained with LM and Eigen vector analysis.

❄ Ophthalmology

Güler and Übeyli [126] reported that Elman-RecNN trained with LM algorithm along with wavelet transform successfully diagnose arterial disorders. The long term Doppler signals from ophthalmic/internal carotid arteries of healthy subjects and patients are decomposed into time–frequency domain with discrete_wavelt_transform in the preprocessing step.

❄ Brain and CNS

On the physiological front, brain-/CNS aberrations, old-age challenges (dementia), psychological disorders (autism) etc. are related directly or indirectly with neocortex. The cross-disciplinary experts center their focus on blue print of microcircuits of neurons in neocortex from physiology, computer modeling, surgery, electronics, and mathematical emulation stand points of view. The staunch hope is to correct microcircuit deviations through surgical intervention for a better health in totality. Neural network (whatever it is!) is a building block of micro as well as mega interplay of life with common-sense/intelligence, known/unknown, pain/melancholy, pleasure/happiness, concern/indifference, comfort/discomfort, ambition/contentment, excitation/calmness etc.

Behavior of brain: Four Rec-NNs viz. regular, random, small-world and scale-free are used to study the architecture of the network. Random network is robust when the noise is high while small world NN has high performance in decision making for low noise. Random and scale free networks are robust to distributed/clustered damage.

Compression of EEG signals: Sriraam [96] studied context-based near-lossless compression of EEG signals using LMS adaptive FIR filter, auto regression (AR) and NNs viz. SLP, MLP, Elman and Gen_Rec_NN. SLP is best when using near lossless compression procedure without considering errors in the model.

Chemical diffusion in cortical brain: Gross and Hanna [87] proposed NN models to enquire into chemical diffusion spatiotemporally in cortical brain. The local function of chemical in the brain of a freely behaving animal using chemical sensor arrays throws light on neurochemical and pharmacological mechanisms responsible for the diseases of the brain (Chart 17).

Chart 17: Tools for chemical transformations

- ❄ Imaging systems
  - + Provides spatiotemporal chemical information
  - — Patient to remain in still posture during recording
- ❄ Implantable chemical sensors
  - + Provide higher resolution than imaging system

- \+ Freely behaving animals
- \- Proximity to the sensor to the animal

❆ NNs

- \+ 3D chemical information
- \+ Learning ion diffusion in a model of cortical brain
- \+ Non-noisy/ noisy conditions

❆ Functional imaging techniques

- ▶ CT, fMRI, PET/SPECT MRS

Neurologically healthy volunteers: Vuckovic and Sepulveda [59] calculated absolute values of Gabor coefficients by independent component analysis, followed by determination of classes with Elman-RecNN. This two-modality-four-class brain-computer-interface classifier is tested with ten right-handed neurologically healthy volunteers with an accuracy of 60-70%.

Epileptic patients: Güler et al. [134] reported Elman RecNN with Lyapunov exponents is superior to FF-NN to classify epileptic patients during seizures and seizure-free period from EEG signals. The NN is trained with LM and this method is trust worthy for early detection of the electroencephalographic changes by the analysis of long-term EEG signals. Jansen and Desai [169] analysed digitized data of EEG with MLP and Elman-RecNN. Pravin Kumar et al. [86] reported wavelet entropy features with Elman Rec and RBF NNs detecting normal vs. epileptic seizures (99.75%) and interictal focal seizures (94.5%).

Schizophrenia: Schizophrenia, a psychological aberration prevalent in almost 1% of people [136] arises due to disturbances in synaptic connections in cortico-cortical neuronal circuits. The auditory hallucinations frequently occur in most of patients suffering from Schizophrenia. It is a consequence of a deterioration of the neural structures responsible for speech processing. The basic accepted reason is developmental disturbances of synaptogenesis and/or synaptic pruning during adolescence.

Sleep apnea: Maali and Al-Jumaily [46] modeled to predict sleep apnea of five subjects' episodes by MLP, RBF and Elman NNs. Area under the curve statistic showed higher performance of MLP.

Dementia: Mahmoud et al. [36] studied long term prediction patterns of patients inhabited in intelligent environment. Occupancy sensors monitored movement patterns of aged people. The data is converted into temporal sequences of activities and modeled with NARX and Elman RecNNs. The simulated and real life data indicate NARX is better than Elman-RecNN for prediction of behavior over long term.

Cognitive dis-inhibition: Stein and Ludik [154] proposed Elman NN to model cognitive dis- inhibition in obsessive-compulsive disorder (OCD) arising due to serotonin/dopamine dysfunction. It is an awesome coherence of psychobiology of OCD and neurotransmitter activity.

❆ Spinal cord injury

Maksimovic and Popovic et al. [159] studied classification of functional movements in 16 spinal cord injury patients and seven healthy control subjects with MLP_BP, RBF, Elman-RecNN, SOM and LVQ.

❆ Intensive care unit (ICU)

Intracranial pressure: A model for patient care in intensive care unit (ICU) is complicated and non-linear in many measureable and/or predictable/ unpredictable physiological processes. Shieh et al. [137] used Rec_NN for intracranial pressure in neurosurgical intensive care unit using end-tidal of carbon dioxide (EtCO2), heart rate (HR), mean arterial pressure (MAP) and regional cerebral oxygenation (rSO2). It is also used in solving a chaotic time series simulated from Mackey–Glass differential-delay equation.

f-MRI: Hsu [198] reported a dynamic Rec-Fuzzy-NN with a structure (architectrue) learning method. The node construction and node-pruning phases dynamically result in optimum NN structure. It is applied in a second order chaotic non-linear system. The data of resting – state in human brain are modeled with AR(1)-CSI (cubic spline Interpolation). It is better than simple AR(1) and ARIMA (4,4) models [285]. It is to be noted estimating the true model of the data in fMRI is difficult.

6.11 Robots

Al-Araji [41] minimized tracking error in presence of an external disturbance of a nonholonomic mobile robot with hybrid system of NNs. Modified_ElmanNN models kinematics /dynamics and identifies the posture. The reference torques controlling steady-state outputs are results of FF-NN. Mbede et al. [133] integrated modified Elman-RecNN and robust controller to account for uncertainties in robot systems in changing and dynamic unstructured environments. Köker [40] employed GA for training Ws of Elman RecNN with error in end-effector position as fitness function for a six-joint Stanford robotic manipulator. A learning_ robot learns by NARX and Elman NN, while a demonstrator robot executes the action. Here NARX is superior to Elman RecNN. Köker [127] reported an intelligent predictive controller for a six-degree-of-freedom robot.

Multi-fingered robotic hand: A Rec. NN is reported based on cone function and another NN using gradient of merit function derived from Fischer-Burmeister_natural residual function. They are successfully applied to simulation of a second order cone problems (which are linear or nonlinear convex) and grasping force optimization in multi-fingered robotic hand. This research concerned with NNs was purely physiological and was targeted to monitor mental health, diagnosis and treatment of brain deceases.

6.12 Finite automata

Sperduti [165] reported that Elman-RecNN simulates any finite state automata as well as multi-stack Turing machine and any frontier-to-root tree automation. But, rec-cascade correlation does not simulate any finite automata.

Transform invariant representation of objects: Rec_NN using association [213] acquires the capability of transforming invariant representation of objects. It is shown that the network stores the object representation from any one of the views and also can retrieve it correctly. Further, the effect of distortion of retrieval and dilation of connectivity on the efficiency of Rec_NN is studied.

6.13 Engineering

Due to the lack of a complete access to the system states in different engineering applications, it is required to estimate these quantities. Therefore, during the past four decades, state estimation of dynamical systems has been an active topic of research in different areas such as: fault detection, monitoring, process control and biomedical systems. Automatic control techniques usually assume complete accessibility for the system states, which is not always possible (cost, technological constraints, etc.). Several approaches consider a nonlinear transformation or a linearization technique. In real applications, there are external disturbances and parameter uncertainties and they are often ignored. Although, robust techniques (observers) have good performance even in the presence of uncertainties, their design is complex.

- **Fault detection**

Automatic fault rectification: It requires a complete data regarding states of systems under normal and malfunctioning scenarios. Since, in biomedical, electrical/mechanical systems, the mapping of all states is impossible/ impracticable due to cost/time, linearization (under simplifying assumptions) or non-linear transformation was in practice. Genetic algorithms for parameter optimization and Takagi-Sugano fuzzy models in the design gained momentum. Rec-NNs are coveted in dealing with NL-systems in discrete/continuous time in presence of uncertainties of measured signals, external process disturbances even in absence of knowledge of model dynamics.

Air handing unit: Fan et al. [84] employed multiphase strategy for sensor fault detection, diagnosis in air handing unit with NNs and wavelet strategies.

Phase 1: The fault detection model consists of two MLP_BP-NNs which are trained with normal operating data of system. The sensitivity analysis is performed for first MLP and it drives the second FF-NN in the same control loop

Phase 2: wavelet analysis outputs approximation coefficients sensors of the measurements. The faults in the sensor are identified by Elman RecNN. The results show successful detection, diagnosis of fixed biases and drifting fault of sensors in system of air handing unit. Cluster information from fuzzy c-means enhances the reliability.

Motor winding: Asfani et al. [55] reported fault detection in induction motor winding. The occurrences arise by transient phenomena at the starting and ending point's short circuit and the current signal is wavelet transformed. The input to NNs is wavelet transformed energy level of high frequency signal. Elman-Rec_NN is better than MLP and RBF NNs which have only feed forward connections.

Transmission lines: Deihimi and Solat [31] compared echo state networks for distance protection compensated by thyristor-controlled series capacitor with Elman/ time-delay/RBF NNs and NARX models. A big_bang–big_crunch algorithm was used for optimization of design parameters of echo state networks. The system is tested with 7680 test cases with varying fault inception-angle, fault resistance, load angle, fault location and compensation degree on a 400 km, 500 kV line.

Machines failure: Liu et al. [122] employed similarity based method and Elman-RecNN to predict long term probabilities of failure of machines in manufacturing site. Elman model is inferior for this task.

Under water vehicle control: Under water vehicle control is complicated requiring self-tuning regulator (STR) and model reference adaptive control. This needs continuous identification of the system. FF-NN and Rec-NNs were implemented in underwater-vehicle for control.

6.14 Aircraft in autopilot mode

The performance of automatic landing system (ALS) is crucial in flight landing and increased safety of aircraft landing. In autopilot mode of aircraft, GPS and INS are integrated (KB. 4). It gives navigation even in absence of GPS. Elman-/Jordan- NNs are trained with GA, PSO and EA algorithms. Juang [272] proposed Rec-NN with GA in ALS to improve safety. RTRL is used in training Rec-NN and five crossover methods (Adewuya, arithmetical, average and convex_blend) of GAs to get optimal control parameters. The performance is better than conventional controllers.

KB. 4: Aircraft in autopilot mode

If GPS +KF
Then Robust solution

- Prior knowledge of error model of INS needed
  Remedy : GPS + NN
- Non-convergence and inaccuracy
  Remedy : Elman, Jordan RecNNs
- BP Training
- Convergence and inaccuracy
  Remedy: EA

Tunnel operations: Guo et al. [26] predicted tunneling induced ground deformation with a combination of wavelet, Elman-RecNN and WIPS models. The ground deformation is decomposed into trend and wave components with wavelet transform. Elman with PSO identifies deformations and prediction with WIPS follows. This hybrid model is viable compared to translation of complicated set of rock-soil processes into mathematical parlance. Their unique accurate solution has a practical utility in subway tunneling operations.

6.15 Electrical power

Electric load prediction: Benaouda et al. [125] studied 1-hour-ahead electricity load in New South Wales (Australia) with hybrid paradigms. The signal is subjected to multiple resolution decomposition using the non-decimated or redundant Haar_trous wavelet transform, which considers asymmetry in TS. The results of Elman-RecNN are compared with AR, multi-scale AR, MLP and GenRec_NN.

Power transmission: The decision of digital distance relay is very important for making the protection scheme in the transmission system more reliable. As the current signal is taken from the output of the current transformer, the distortion introduced by the saturation in it affects the performance of the distance relay. In this context proper nonlinear modeling of current transformer is necessary and a suitable compensation should be carried out to nullify the distortion introduced by it. Temurtas et al. [138] reported FF_NN, Elman_Rec_Nn to detect every (5th, 7th, 11th, 13^{th}) harmonic in active filter. A distorted wave from power line is analysed by Fourier transform. The fundamental wave is removed through a low pass and then harmonic recognition follows.

Electric bus vehicle: Hybrid electric buses technology is environmental friendly in emission of lower amount of CO2 and at the same time has lower fuel consumption. Wang et al. [51] investigated a two phase optimization of a suboptimal energy management strategy to control a series–parallel hybrid electric bus vehicle in real-time (chart 18). The first phase consists of reaching an optimal energy_ management_ strategy with iterative dynamic programming for bi-objective cost function. It is followed by Elman-RecNN to arrive at a sub-optimal phase implemented in vehicle control unit of the hybrid bus for a real ride on the high way. Extensive studies are performed on a hardware-in-the-loop simulation system constructed on PT-LABCAR. Here, a virtual system of vehicle, driver and driving environment is considered for forward-facing HEVs.

Chart 18: Object functions in electric bus energy management

- ObjFns:
  - Min(fuel consumption)
  - Zero battery state-of-charge change
  - Avoiding frequent clutch operation

Automobile wiper: Zolfagharian et al. [29] made use of NARX-Elman-NN in minimizing unwanted noise and vibration of automobile wiper blade employing experimental data during its operation. A bi-level adaptive-FIS with multi objective-GA deals with conflictive interests in this complicated functional module. Zolfagharian et al. [42] introduced a multi-objective control strategy for automobile wiper blade to function within its sweep workspace for a small amount of time with noise and vibration optimum. The first step is collecting noise and vibrations when the wiper is on. In the next step, NARMAX and Elman-RecNN developed black box system identification models. The third phase consists of closed loop iterative controller with Pareto_ multi_objective_GA for the wiper system.

6.16 Mobile communications systems

Rec_NNs are used in mobile communication systems extensively. Adaptive equalization is an important activity in mobile communication. MLP was used as an alternative to its linear transversal equalizers where the transmission channel is time dispersive and non-time varying. MLP was found superior to LTE as the latter is limited by the optimal linear Weiner solution in adoptive filtering involving equalization of time varying fading channels. FF_NNs had overwhelming success compared to conventional techniques Volterra series, RBF and Rec_NNs etc. in use for equalization tasks. Gao et al. [152] put forward multi-step ahead prediction of the occurrence of long term deep fading in the mobile communications systems. The modified Elman NN using temporal difference is employed.

Rec-NN for SDMA: It has 2*n external inputs and Kv outputs. Rec-NN is trained with sequences transmitted by desired users (k0) and the purpose is to separate multiple end users [218]. The output enables to place the beam of antenna in the direction of desired users while the nulls are focused in the direction of interfering users. RTRL-online algorithm in training has an additional application in tracking the movement of the mobile user. A simulated system with six receiving antennas and six users is implemented using two structures. Each structure requires 1125 multiplications, 585 additions and 3 sigmoid functions,but demodulates three users only. The performance is superior to CDMA under high values of signal to interference ratio.

6.17 Pattern recognition

Pattern recognition is the most subtle, complicated task even with today's information technology. Typical studies in this decade include identification of digit '8' and characters ('T' and 'C'), prediction of subsequent symbol in a continuum of stream of inputs, sequence [235], circle-generation of color textures, analysis of fuzzy grammar, embedded Ruban grammars and rule extraction. In the dissolution of drugs (pharmaceutical preparation) and classification of cervical cells (biochemical/histo-chemical studies), the outcome of Rec_NNs is noteworthy

Prediction of a sequence of ordered points: A series of 12 ordered points giving the shape '8' is trained by Rec-NN. The prediction of the next point is not possible with a static NN because pointed co-ordinates (0,0) have two successive points 5 and 11. Rec-NN decides the successor based on its predecessors. For example if the predecessor are 3 and 9, then the successors are 5 and 11.

Archtec.Rec-NN: The architecture consists of two input neurons for which the two point co-ordinates are the input. The two co-ordinates of the predictive point are in the two output neurons. There are two hidden layers with four neurons.

Eigen values: A continuous Rec-NN is used [200] to compute largest and smallest Eigen values of a symmetric positive pair (A, B). But oscillatory dynamics of NNs [203] is considered less important as far as information processing is concerned

- **Speech recognition**

Rec_NNs have been successful in speech recognition from utterances by male/female speakers, data in under-water-vehicle and mobile communications. The speaker independent and dependent speech recognition are two tasks solved by statistical as well as NN-methods. MLP-NN was used as a normalization module with consistent reduction of 'word-error-rate'. The sequential nature of the acoustic feature vectors requires the changes with time sequence and hence dynamic techniques need to be adopted. Speaker independent system was trained using 2140 utterances from 50 male and 50 female speakers from APASCI, Italian speech data base recorded in a quiet room. For test data, four speakers are considered. The utterances of each one lasts for 12 seconds on average corresponding to 19 volts. In the adaptation and test phases, the speaker makes 50 and 30 utterances on different days in an office environment. The results are with a great success in reducing WER from 35.6% to 15.4% in case of speaker dependent speech recognition system. . Amrouche et al. [90] probed into efficacy of MLP, Elman, HMM models in pattern recognition of Arabic digits under different noise scenarios viz. SNR variation, multi-speakers, babble background, car production hall (factory), military vehicle (leopard tank) and fighter jet cockpit (buccaneer). These noise patterns are procured from NOISEX-92 database. DeLiang et al. [167] modeled speech data analysis with Elman RecNN.

- **Linguistics**

Recognition of hand writing, grammatical inference and speech recognition have long term dependency i.e. the output depends on inputs occurred long ago. In short time scales, the sequences are characterized by the dynamics. Regular language inference requires the state information to be stored over indefinite period of time and no feature extraction is necessary for learning. Syntactic and grammatical structures are relevant for long term time scales. The grammatical inference belongs to long term sequence while speech recognition involves short term phonetic features as well.

Prediction of next-symbol data: Čerňanský et al. [118] mapped the prediction of next-symbol data sets containing recursive linguistic structures into Elman-RecNN. The increasing of even small depths of recursions mimics human performances at least partially. At the start of training, clusters of activations of RecNN are like Markov chains and the small random weights resemble Variable Memory Length Markov Models. As the training proceeds, the state-space of NN is reorganized depending upon categories of words and grammatical substrings. After optimum training, the prediction is according to rules of grammar also rather than simple individual words. Cartling et al. [103] elucidated implicit acquisition of grammar (context free) by Elman-Rec and MLP- NNs. PCA of hidden layer activities showed that there is well organized internal representation of grammatical elements. Omlin [205] trained Rec-NNs which predicted the next symbol using the truncation of backward recurrence. The hidden unit activations represented past histories and clusters of their activations represent the states of the generating automation. The complete deterministic finite state automata are extracted from Rec-NNs. The capacity of Rec-NNs in representing symbolic knowledge is tested with grammatical inference systems. In fact, language processing is a good test bed for Rec-NNs. A discrete second order Rec-NN is trained to recognize strings of a regular language from a set of positive and negative examples. The Rec-NN has N-recurrent hidden neurons. The complexity of NN grows as $O(n^2)$, if the number of input is small compared to the number of hidden neurons. The values of hidden neuron collectively are referred as a state vector, S in the finite N dimensional space $[0,1]^n$. Each input string is encoded into the input neuron one character per discrete time step.

Grammatical inference: Chandra et al. [343] proposed co-operative co-evolution of Rec-NN for grammatical inference tasks. It employs evolutionary algorithms to solve a high dimensional search problem by decomposing it into low dimensional subcomponents. The prospects of evolving both Ws and the network topology await attention. The use of different coding schemes during training results in optimal systems [222]. Rec_NNs model crisp grammatical inference systems from positive and negative examples. Rec_NN is trained with real coded GA for inference of fuzzy grammar.

❖ Commerce

Xue and Keet [111] proposed a hybrid system of Elman RecNN and rough sets to predict five-category risk grades in financial data of 896 firms. This model excels logistic model.

Forex data: Bildirici et al. [78] tested, TAR-VEC-RBF, TAR-VEC-Rec-Elman with datasets of monthly returns of TL/$ real exchange rate and ISE100 Istanbul Stock Exchange Index. The order of efficiency is TAR-VEC-Rec-Elman >TAR-VEC-RBF > TAR-VEC-MLP. For long run prediction RBF is the best.

Financial forecast: Huang et al. [132] reported SVM and its hybrids outperformed LDA, QDA, Elman-RecNN in predicting direction of movement financial sector.
Tourism development: Cho [146] reported Elman-RecNN excelled exponential smoothing, univariate ARIMA in prediction of travel demand (i.e. the number of arrivals) of Hong Kong from different countries.

7. Theoretical results

Abbreviations		Definition
LMI	:	Linear matrix inequality
TS	:	Time series
Globexp	:	Global exponential

Time delays due to integration and communications are ubiquitous in biological neural nets. In fact, this is a source of instability with a lot of beneficial outcome. In the mathematical front, the stability analysis of RecNNs (in general any NN) is necessary from theoretical stand point and activity goes on searching for existence and uniqueness of equilibrium points and varieties of stability viz. global, asymptotic, robust etc. at these points. In the case of globally recurrent mathematical NNs (MaNN), the stability is hard to be proved. For, Recurrent_NNs oscillatory dynamics is difficult due to the fact the mathematical tools for the analysis of periodic orbits in high dimensional spaces are not available. However, locally Rec_NNs permit easy checking of the stability by the examination of poles of their internal filters. Hence, the viable approach is parameter studies of discrete neuro dynamics. Li at al. [418] proposed criteria on stable/semi-stable/ positive_semi-global_stable/unstable continuous attractors with infinite neurons in higher-order Rec_NNs using linearization technique. The simulation results under Log-normal distribution supported the developed theoretical propositions. The necessary and sufficient conditions for the existence of a stationary solution for the multiple class Rec_NNs are derived. Zhang et al. [286] conducted stability analysis for a class of discrete-time-stochastic-delayed-NN with parameter uncertainties. The distribution probability of time delay is translated into parameter matrices of the transformed NN model. The stochastic disturbances are described in terms of Brownian motion. Time varying delay is characterized with Bernoulli stochastic variable. The Rec-spiking NN [304] based on LSM leads to convergence of Ws using spike time dependent plasticity. The general practices of the protocol of stability analysis are described in Alg.6.

Alg. 6: Stability analysis protocol

Step : 1 Translate equilibrium of NN to a zero solution
Step : 2 Stability analysis of it around zero

If Equilibrium exists
Then Stability of NN = stability of zero solution of transferred system
Real life tasks : Dynamic behavior of NN =
fn(External stimulae, parameters, external stimulae)

— Translation cancels all external stimulate
Consequence: conditions stability are independent of external stimulae
— It is over conservation

Now, probing into stability of Rec_NN with delays is current focus and typical directions of research during last two decades are documented in Table 9.

Table 9: Current trends in stability analysis of Rec_NNs

$$$-	Task		Rec_NN-characteristics	Ref
Converge to equilibrium points	Mimicking human's memory patterns	Memristor-based	Lyapunov Differential inclusions theory	411
Global exponential	Robustness analysis		I#:time delays random disturbances	411
Global exponential		Delayed RecNN Chaotic	Decoupling technique LMI	382
Global exponential	Mixed discrete and distributed delays Existence and uniqueness of the equilibrium point under mild conditions, assuming neither differentiability nor strict monotonicity for the ActFn.			240
Globally asymptotically stable in mean square	Stochastic discrete-RecNN		Lyapunov–krasovskii function linear matrix inequality (LMI)	406
Global exponential in	continuous RecNN			349

Table 9: Current trends in stability analysis of Rec_NNs

$$$-	Task		Rec_NN-characteristics	Ref
Lagrange sense	multiple time delays			
Global robust exponential	equilibrium solution to delayed reaction–diffusion RecNNs with Dirichlet boundary conditions on time scales		Topological degree theory m-matrix lyapunov inequality skills	332
Asymptotic			Lyapunov	305
Exponential		Stochastic memristor-based-RecNN	Time-varying delays	220
Global exponential	Synchronization	Memristor-based RecNN based	Fuzzy theory Lyapunov time-varying delays	389
	Any neural state is globally convergent to the feasible region in finite time and stays there thereafter. If objective function and constraint functions are pseudo convex Then any neural state is globally convergent to the unique optimal solution, If constrained invex optimization problems & penalty parameter is sufficiently large Then any state of the proposed neural network is globally convergent to the optimal solution	One-layer rnn		426
Global asymptotic	Stochastic-RecNN Multiple discrete time-varying delays and distributed delays		Lyapunov–krasovskii LMI	283
Ensure existence, uniqueness, global exponential global convergence	ActFn - discontinuous		Image-matrix, lyapunov-like approach, Des with discontinuous right-hand side as introduced by filippov.	300
Multi	Coexistence of stable and unstable equilibrium points			299
Global exponential	dissipativity	Memristor-based- RecNN Time-varying delays	Lyapunov image-matrix Lasalle invariant principle	384
Solubility and	Deterministic RecNN noisy TS			347
Global exponential in Lagrange sense	Continuous RecNNs with multiple time delays.	ActFns bounded and unbounded		265
Sure exponential , mean	convergence dynamics of reaction–diffusion RecNNs with continuously distributed delays and stochastic influence		Lyapunov m-matrix nonnegative semi-martingale convergence t	195
			Lyapunov stability	358
Exponential	Hybrid stochastic RecNN		Razumikhin-type	262

Table 9: Current trends in stability analysis of Rec_NNs

$$$-	Task		Rec_NN-characteristics	Ref
exponentially convergent rate			theorem	
Imageth moment exponential	stochastic RecNN Time-varying delays		Method of variation parameter and inequality techniques	245
Hopf bifurcation analysis	Distributed delays Strong kernel			327
Global exponential	Design with piecewise constant argument of generalized type		Lyapunov	324
Convergence analysis	Discrete time RecNN multivalued neurons (mvn), complex-valued weights activation function defined as a function of the argument of a weighted su.			298
Exponential	Both time-varying delays general ActFns	Not considered boundedness monotony on these ActFns Differentiability on the time-varying delays	Lyapunov functional free-weighting matrix	268
Global			Lyapunov	328
Global convergence				291
Multi-		n-d_ complex-valued RecNN ActFn : real-imaginary- .		
and convergence	modified Hopfield model			261
analysis	Discrete-time RecNN time-varying delay		Linear matrix inequalities (LMIs)	295
		Augmented Lyapunov–Krasovskii functional & derivative	Time-varying delay	206
Global exponential	Discrete-time-RecNN Impulses			309
analysis	discrete-time RecNN With time-varying delays		DiscreteJensen inequality and the sector bound conditions, (LMIs)	296
Attraction domain	RecNN time-varying delays		Lyapunov–krasovskii functional lyapunov–razumikhin functional method Invariant set principle	266
Imageth moment exponential	Stochastic RecNN Unbounded distributed delays			322
The imageth moment exponential and imageth moment global asymptotic	Stochastic functional differential equations with infinite delay.		Razumikhin method and lyapunov functions	
Attraction and exponential of the almost periodic solution	To design globally stable almost periodic oscillatory nns		Lyapunov	302
Global asymptotic	Stochastic RecNN Time varying delays		Lyapunov LMI	277
	Optimization task Globally convergent Exact optimal solution	Lyapunov,	One layer-RecNN	404

Table 9: Current trends in stability analysis of Rec_NNs

$$$-	Task		Rec_NN-characteristics	Ref
		Time-varying delay interval	LMI	386
	Mixed RecNN	Lyapunov .	LMI discrete and distributed delays	402
Global exponential	Robustness	Stochastic-RecNN		385
Exponential	delayed RecNN		Lyapunov–krasvoskii functional delay partitioning method	412
	Static RecNN	Lyapunov– Double/triple integral	LMI	380
Global exponential	Delayed RecNN Parameter uncertainty in connection weight matrices			238
Exponential convergence		memristor-based RecNN	Theory of ODEs with discontinuous right-hand sides	356
Global exponential periodicity		Memristor-based RecNN	Lyapunov functional	409
Global exponential			Mixed delays	416
	Time-varying delay-dependent	Lyapunov	LMI inequalities	419
dynamica l	Multiple equilibrium points in RecNN Time-varying delays ActFn discontinuous			364
Global exponential	Robustness Parameter uncertainty in connection weight matrix.			398
Exponential	Time-varying delays	Memristor-based RecNN		365
analysis	stochastic-RecNN	Mixed delays Markovian		392
~~Xxx~~				415
Bifurcation analysis	Three-node RecNN Four discrete time delays		Lyapunov–krasovskii functional + (LMI) Integral inequality approach (iia)	397
Robust	Time-varying delay-dependent conditions			396
Global robust exponential stochastic	Discrete-time uncertain-RecNN uncertain parameters markovian jumping Time-varying delays	LMIs		335
Training	Nonmonotone BFGS self-scaling BFGS + [adaptive nonmonotone technique + approximations of Lipschitz constant]	>> bfgs		348
Exponential	Imageth moment	Stochastic delayed RecNN		417
analysis exponential	Stochastic RecNN Unbounded time-varying delays	pth moment Lyapunov function semi-martingale convergence	M-matrix technique	353

Table 9: Current trends in stability analysis of Rec_NNs

$$$-	Task		Rec_NN-characteristics	Ref
Globally exponential of	RecNNs critical conditions		Matrix measure theory	363
Global exponential	Time-varying delays Differential equations with discontinuous right-hand side as introduced by Filippov	Memristor-based		361
Global-	Optimization Convergent to an exact optimal solution	Lyapunov		425
Global exponential	Impulsive delayed dynamical systems generic criteria	Extended halanay differential inequality		342
Global exponential periodicity Global exponential	Various ActFns and time-varying delays ActFns Monotone nondecreasing Globally lipschitz continuous Monotone nondecreasing Semi-lipschitz continuous Mixed monotone functions, lipschitz continuous function			243
Global asymptotic	Time-invariant delay Delay-dependent stability criteria Static-RecNN	Lyapunov–krasovskii functional		297
Asymptotic	RecNN	P-critical conditions, i.e., a discriminant matrix image is nonnegative definite, where image is a matrix related with the network and p is an arbitrary nonnegative definite matrix.		317
Existence, uniqueness	Time delay in the leakage term under impulsive perturbations	Do not require Boundedness, differentiability monotonicity of ActFn checked with linear matrix inequality (LMI) toolbox in matlab		323
Global exponential	Discrete-time-RecNN			344

8. Future scope

The neocortical microcircuits in human brain computes and the throughput excels all known technologies of today. A promising look at probing into operational principles (mode of computation, parallelism, fool proof results in spite of mistakes through ensembling, plasticity, repetition, integration, differencing etc. …) with available scientific technology opens flood gates for new neuromorphic products. The outcome of proof-of-principle studies is promising future expected tools will be far superior to the silicon based devices of the day.

The realization of partial rat brain on a computer, experimental evidence of boson (fundamental of universe/ ultimate particle of the universe), cancer-proof- naked mole rat, robotic surgery, soccer game playing robots are only testimony of scientific pursuits results valid at six-sigma-level repeatability. The warning ‘available (human) intelligent tools are able to do the job of plucking low-hanging fruits’ waves away the pride that the best of the best is grabbed. It reorients the target towards knowing the truth value/ falsehood of known_unknowns, known_knowns, unknown_unknowns and unknown_known in lifeless to life, elementary particle to mega_structures through large (bio) molecules, effects of very high to ultra-low values of temperature, pressure, gravity, volume and electric/magnetic fields.

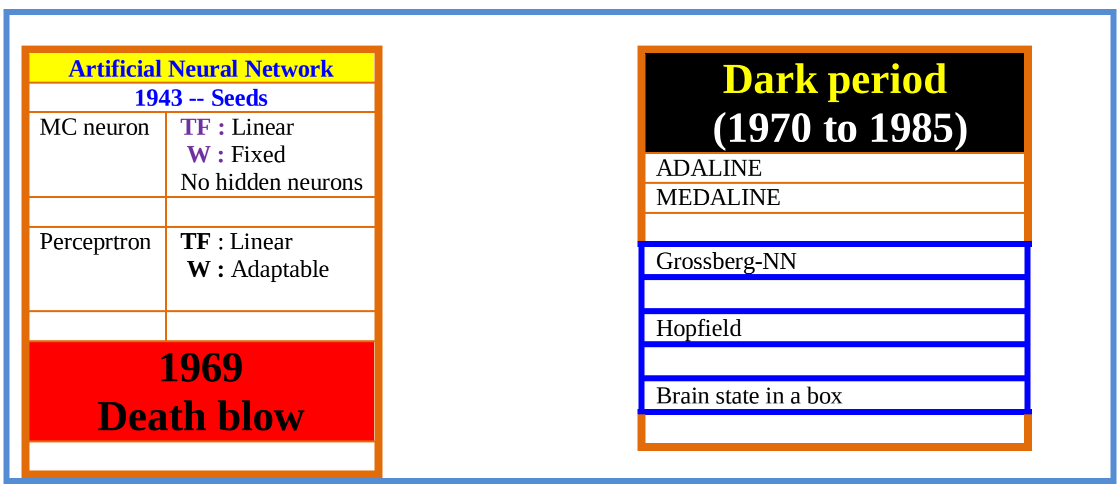
The current Rec-NNs are not guaranteed for the optimal solutions to NP-hard combinatorial optimization problems. Further, there is difficulty in solving non-convex optimization tasks. The future investigations will be around in depth analysis of the dynamics of Rec_NNs for solving non-convex and discrete optimization tasks. On the pragmatic front, thrust area is design of recurrent_NNs for real-time dynamic large data sets. Hardware implementation of these NNs will open new vistas in the instrumentation applicable in field studies.

In future ventures also, the cyclic verdict 'ten years ago, this task was intractable, but now it is too tiny to pay attention' is a gold standard for researching discovery of knowledge. A lightning hope is the thousand fold increase of computing power and smart algorithms by 2020. It promotes realization of a new computing paradigm to augment experimental observations, making predictions of unknown physics/biology in an attempt to be nearer to true nature of real nature.

ACKNOWLEDGEMENTS

I behold with profound reverence to my maternal uncle (Tenugu pandit, sanscritist and asu kavi) who taught me how to learn lessons in my high school days. He asked me to follow the practice; 'read a paragraph sentence by sentence a few (4 to 6) times; close the book and retrieve it; read second paragraph and retrieve it; then try to retrieve both paragraphs. If some sentences are not remembered, read them again. Repeat it for at least for two/three foolscap size pages or two hours of time. Next day without going through material, try to retrieve. If few sentences are out of memory, memorize them again for the next two to three days. The retrieval process is to be tried after a week preferably. It will be almost in memory and can be retrieved with less effort afterwards. A continuous revision for few hours has a consequence of more than eighty to ninety percent error free reproduction.

Appendix A1.a: Life cycle of NNs



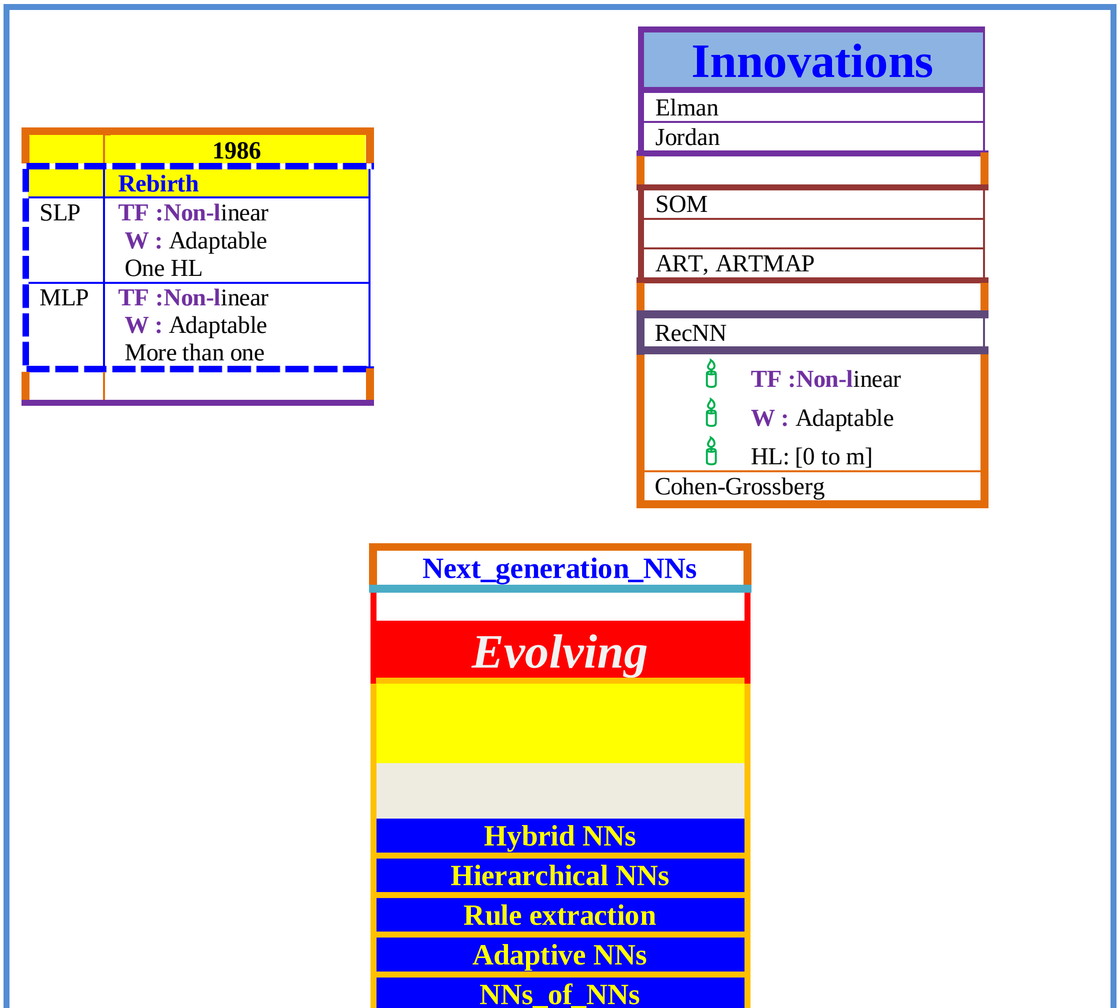


Chart A1.1: Life cycle of NN research (1943-2014- …)

Appendix A1.b: Neurons (Processing Elements)

❄ Biological neuron

Biological neuron is a unit consisting of cell body, dendrite and axon. Dendrite is a structure receiving information from other neurons/bundle of neurons and cell body provides energy. The neural impulse is transmitted through axon. The human brain consists of about 10^{11} neurons and approximately every neuron is connected to 10000 other neurons and neural nets. In essence, it is multi-cellar, biochemical system with non-linear dynamic processes with ion-channels at molecular level.

The system consists of several branches like a tree or nervous system. The ion channels, several in number in each segment of branch control information processing and also responsible its' for passage. The genes in neurons produce messenger RNA which plays a role in production of thousands of proteins. They in turn involve in millions of chemical/biochemical energy transfer interactions. The distributions of these proteins all over neuron (in a need based manner) dictate how memory and information processes take place. The synapses are critical in their function; some transmit signal, others transform in a non-linear manner. The glial cells have a role, but partially understood. The number of neurons, their interconnections is a macroscopic view. But, the concentration changes of ions, blood flow, medicines/neurotoxins/drugs in healthy and those suffering from disorders, biophysics of protein/ion interactions all have their share in total functional aspect of brain. The artificial MC_neuron proposed for a utopian goal of artificial brain celebrated its platinum jubilee. The mathematical transformation and solving differential equations in billions to trillions is one way of mimicking the bio-transformations. The efforts brought renaissance in interdisciplinary knowledge, but still focus remains to be nearer to the realization artificial human brain with today's all evolved technology.

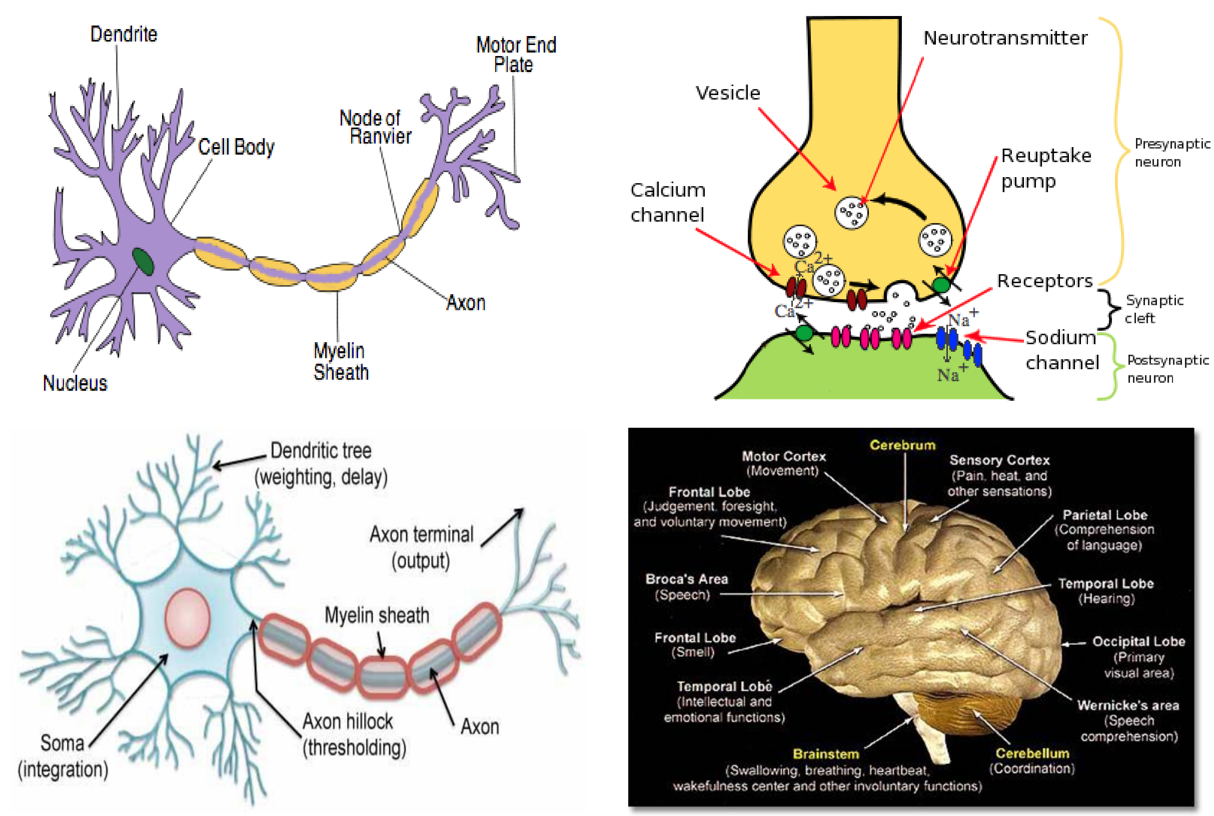


Fig. A1.1a: Biological neuron and human brain

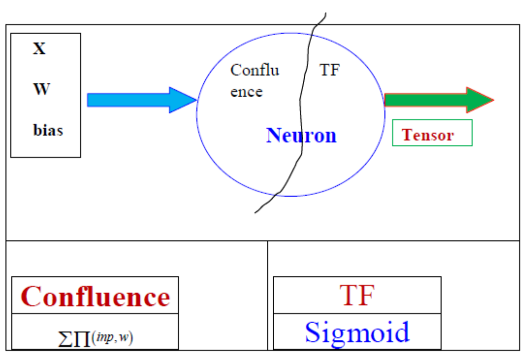


Fig. A1.1b: Artificial neuron (PE)

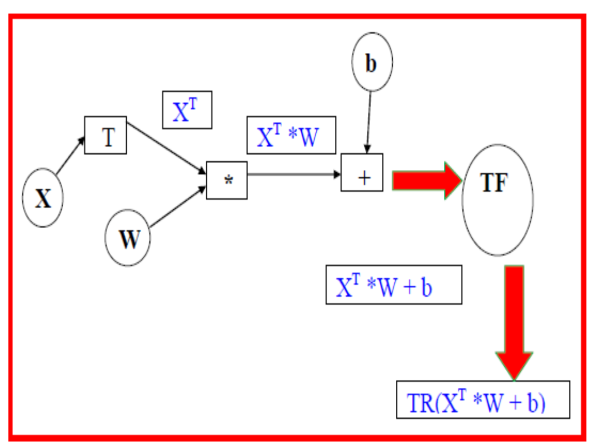


Fig. A1.2: Data flow in Neuron (PE)

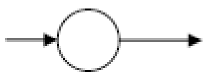
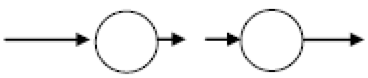
Neuron(s)	Connection tensor	Features
	0 0|[]	▶ Zero neurons
	1 1|0	▶ Single neuron ▶ No feedback connection
	1 2 1|0 0 2|0 0	▶ Two neurons ▶ Not connected ▶ No feedback connection

Fig. A1.3: Neurons not connected

Chart A1.2: Nomenclature in FF-NNs and RecNNs

Layer	
Visible	IL
	OL
Visible	Hidden

Number of neurons	
Layer	
#IL	dimX
#OL	dimL
#HL	

Weights		
Forward	Hidden to Hidden	
WIH	WRH2H	Self feedback
WHO	Output to hidden	
	WRO2H	
Self feedba		

Appendix A1.c: Layered NNs

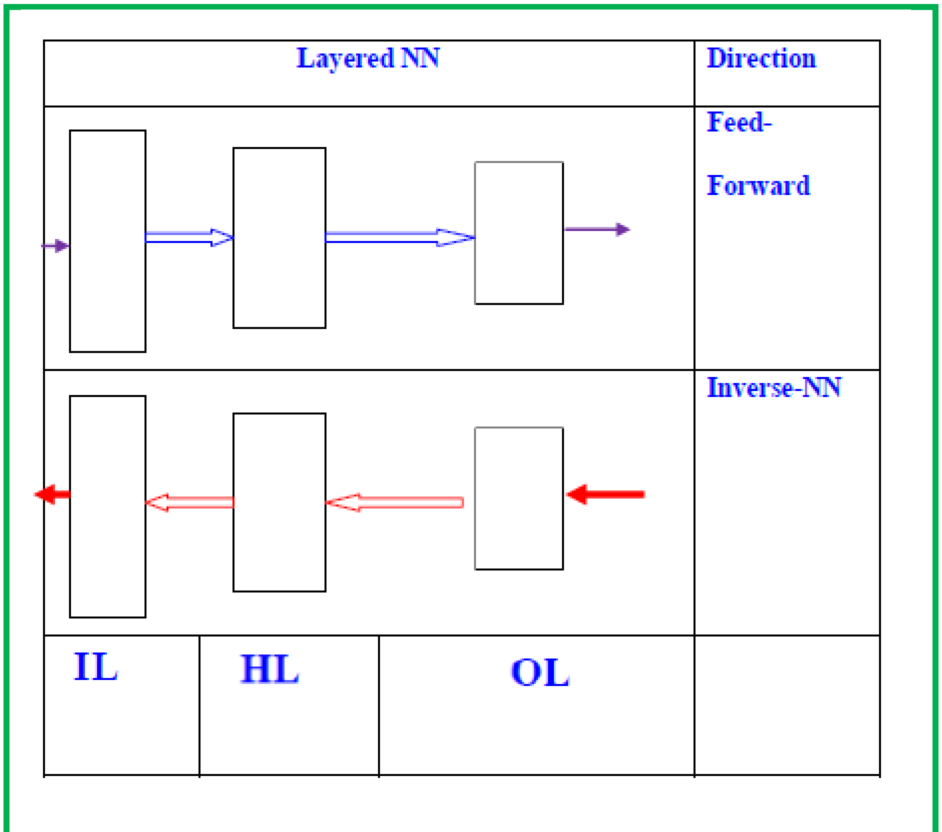
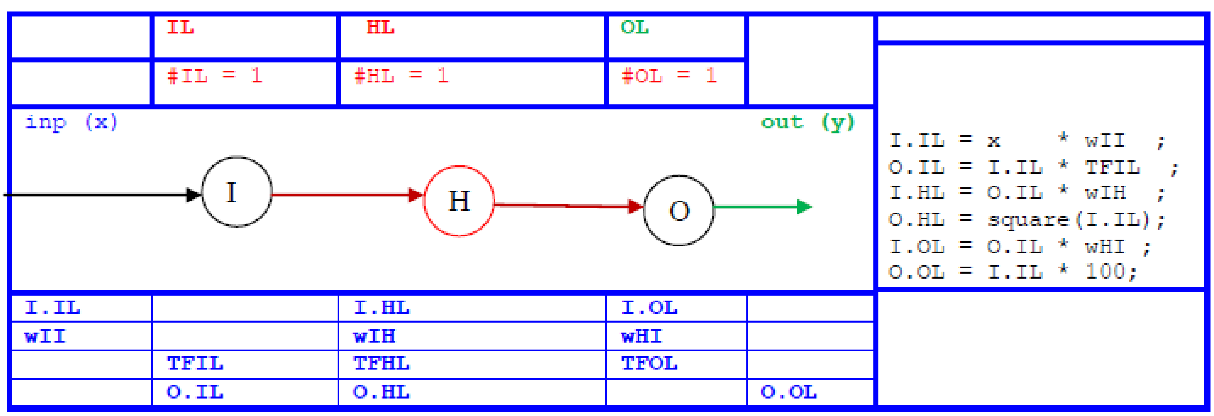


Fig. A1.4: Feed forward (FF) and inverse-FF SLPs



$$x \quad = scal(inp.IL)$$

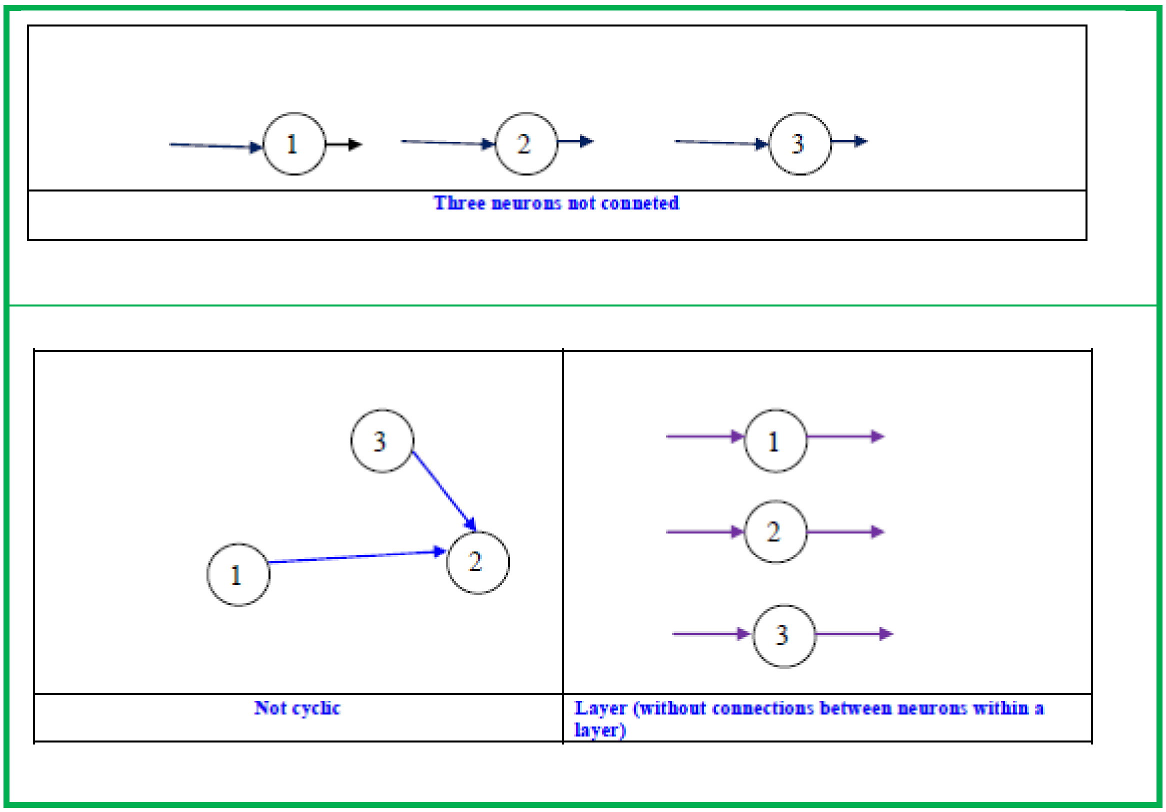
$$O.OL = TF.OL\left[Unscal\left[confl.OL\left\{TF.HL\left[confl.HL\left(TF.IL\left(confl.HL\left\{scal\left(x^{T}\right)\right\},WI\right)\right)\right]\right\},WHO\right]\right]$$

$$y \quad = unscal(O.OL)$$

wt = wII: 1 wIH: 0.1000 wHI: 1	TF = TFIL: 1 TFHL: 'square' TFOL: '*100'	Confl = TFIL: 'dotProduct' TFHL: 'dotProduct' TFOL: 'dotProduct'	
x = 2	I = IL: 2 HL: 0.2000 OL: 2	O = IL: 2 HL: 1 OL: 200	y = 200

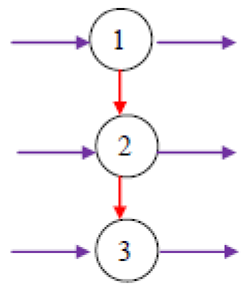
Fig. A1.5: SLP (1-1-1)–FF-NN with three neurons

Appendix A1.d: Non-layered architectures

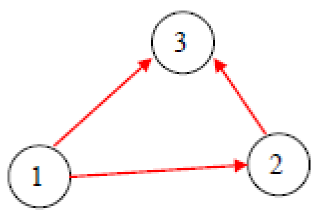


Direction	#connections				
	Acyclic			cyclic	
Forward				3	1,2 2,3 1,3
→Successive	2	1,2 2,3			
→Lateral	1	1,3			
Backward				3	2,1 3,2 3,1
→Successive	2	2,1 3,2			
→Lateral	1	3,1			
Self-feedback	3	1,1 2,2 3,3		3	1,1 2,2 3,3
Total	9			9	

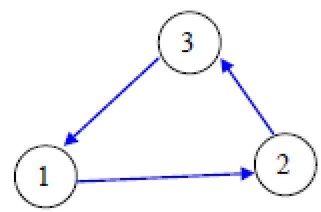
Number of connections and neurons involved



Layer (with connections between neurons within a layer)

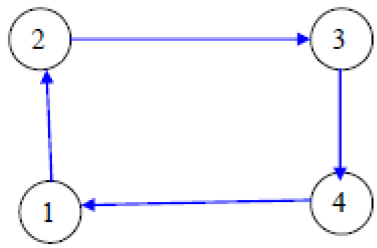


Closed

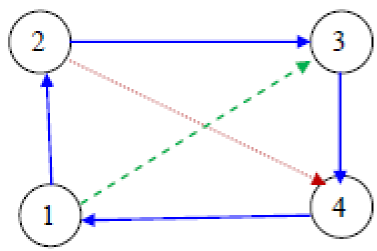


Recirculation

Fig. A1.6: Three neurons closed_loop_NN with no_self_feedback_connections

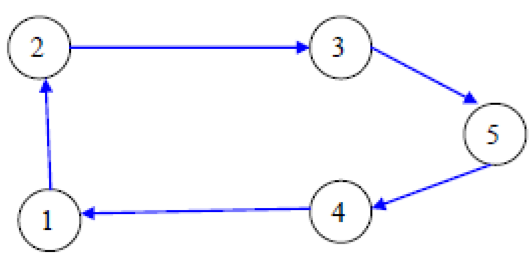


Recirculation



Direction	Acyclic			cyclic		
Forward				4	1,2 2,3 3,4 4,1	
→Successive	3	1,2 2,3 3,4		4	1,3 2,4 3,1 4,2	Diagonal
→Lateral	3	1,3 1,4 2,4				
Backward				4	2,1 3,2 4,3 1,4	
→Successive	3	2,1 3,2 4,3				
→Lateral	3	3,1 4,1 4,2				
Self-feedback	4	1,1 2,2 3,3 4,4		4	1,1 2,2 3,3 4,4	
Total	16			16		

Fig. A1.7: Four neurons closed-loop_NN with no_self_feedback_connections



Five neurons

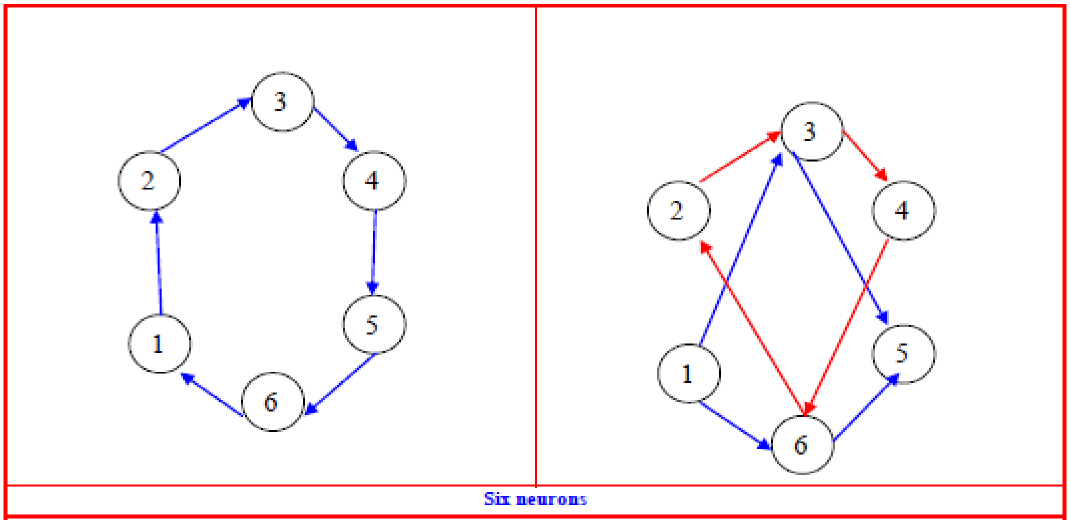


Fig. A1.8: Five & six neurons closed-loop_NN with no_self_feedback_connections

.Appendix A2.a: New generation Neurons (NGN) (or Processing Elements)
Knowledge Intelligence Natural (e') Discovery (KIND)

Quantum neuron

A quantum neuron can have the value 0, 1 or both simultaneously. Thus it imbibes binary bit with an additional feature. A qubit is constructed with hydrogen atom, which consists of a nucleus and one orbiting electron. For the purposes of quantum computing, only the orbiting electron is important. This electron can exist in different energy levels, or orbit(al)s. The different energy levels would be used to represent the binary 0 and 1. When the atom is in its lowest orbit (ground state), it represents the value 0. The next highest orbit would represent a value 1.

NOT gate with Qubit: The electron can be moved to different orbits by subjecting the electron to a pulse of polarized laser light. This has the effect of adding photons into the system. So, to flip a bit from 0 to 1, enough light is added to move the electron one orbit up. To bring it (flip) back from 1 to 0, the same thing is done, since overloading the electron will cause the electron to return to its ground state. This is logically equivalent to a NOT gate. Similarly, AND, COPY etc. systems are constructed. Up to this point, qubit and binary bit function in the same way.

Super position: If only half of the light necessary to move an electron is added, the electron will occupy both orbits simultaneously. Superposition allows two possibilities to be computed at once. Considering a "qubyte", i.e. 8 qubits, represents simultaneously 256 distinguishable states. Then the algorithm is performed on these qubits. When the algorithm is complete, the superposition is collapsed. This results in the true answer. This is where the scaling up of qubit comes into operation. The advantage is running of algorithm on all possible combinations of the definite qubit states (i.e. 0 and 1) are parallel.

For example, factoring a 250 digit number would take approximately 800,000 years on 1400 present day Von_Neumann_computers working in parallel. But, with a Quantum computer, just a few million steps are sufficient. The key element is that using the parallel properties of superposition, all possibilities can be computed simultaneously.

Quantum neuron
— Truth of Church-Turing thesis for all quantum computers is in some doubt

Multiplicative-neuron
In this neuron, the confluence operator is a product of input element and corresponding weight.

SLP-NN with a multiplicative neuron
Yadav et al. (2007) proposed SLP with a multiplicative neuron in the hidden layer with a success for forecasting of TS data. Since lagged responses also play a dominant role in most of auto-correlated time series tasks, recurrent-(multiplicative-neuron)-SLP-NN was proposed (Rec.multi_neuron.SLP).

	Neuron_ multiplicative			
input	**Confl**	**TF**	**out**	**output**
$\rightrightarrows$	$\sum\prod inp, w$	Linear	scalar	$\longrightarrow$

$$prod\ \ x.{*}w$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} . * \begin{bmatrix} 1 \\ 2 \\ 0.5 \end{bmatrix} = x_1 \ * 2 * x_2 \ * 0.5 * x_3$$

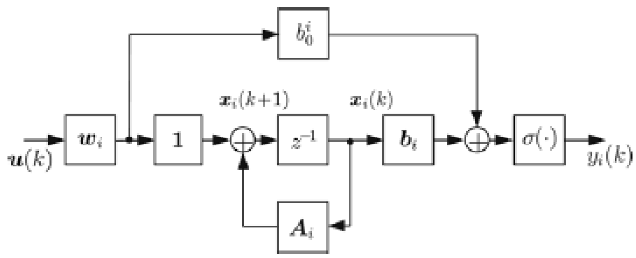
MLP- /SLP-NN
- Non-linear mapping of weighted input
- Models accurate temporal dynamics of chemical/biochemical networks
- Does not require normality or linearity assumptions

- Number of neurons in HLs
  - **Remedy:** Multiplicative-neuron-SLP
    - Only one neuron in HL
    - No a priori knowledge or determining number of neurons
    - Feed-back of lagged responses not considered
  - **Remedy:** Rec-multiplicative-neuron)-SLP-NN
    - Accurate forecast of non-linear TS

Chart A2.1: Neurons

Second-order-spiking-neuron
It is biologically more plausible. SLP with linear TF fails to separate non-linear clusters in XOR. But, an artificial processing (neuron) unit with second-order-spiking characteristics solves XOR classification and simulated trajectory of an arm movement tasks. Here, the inclusion of second order statistics offers significant advantage. A tradeoff between out-put bias and out-put variance occurs by altering the penalty factor in error function.

IIR_ neuron
In this neuron model, IIR (Infinite input Response) or FIR (Finite input Response) filter is introduced between confluence (accumulation) operation and transfer function (TF) of the neuron. It reproduces its own past inputs and activations using input u(t) and output y(t) in the time series mode or u(k) or y(k) in state-space model. The neural networks with these dynamic neurons in the hidden layer appear as if they are of feed forward type (SLP or MLP) at a first glance. But, the functioning with dynamic characteristics is exactly like recurrent NNs, as there is recurrence at neuron level itself. Each neuron of two-hidden layer NN is an rth order IIR filter.



IIR-neuron

- Non-linear approximation is limited
- Remedy: RecNN with two hidden layers

$$x(k)=\left[x^1(k) \quad x^2(k)\right]^T \qquad \begin{array}{l} x^1(k): \text{states of 1st layer} \\ x^2(k): \text{states of } 2^{nd} \text{ layer} \end{array}$$

Rec-NN/41Neural Networks 21 (2008) 59–64

Fig **A2.1**: Rec_NN with IIR neurons.

Appendix A2.b: Hybrid neurons

Wavelet + sigmoid neuron: The addition and multiplication of wavelet and sigmoid activation functions is the core of this hybrid neuron. This output is the input to hidden layer of SLP. It results in summation wavelet neural network and multiplication wavelet neural network. Here, different types of wavelet functions are listed for datasets with dynamic structure. The stability analysis with Lyapunov functions is performed and there is a guarantee of convergence of the learning process.

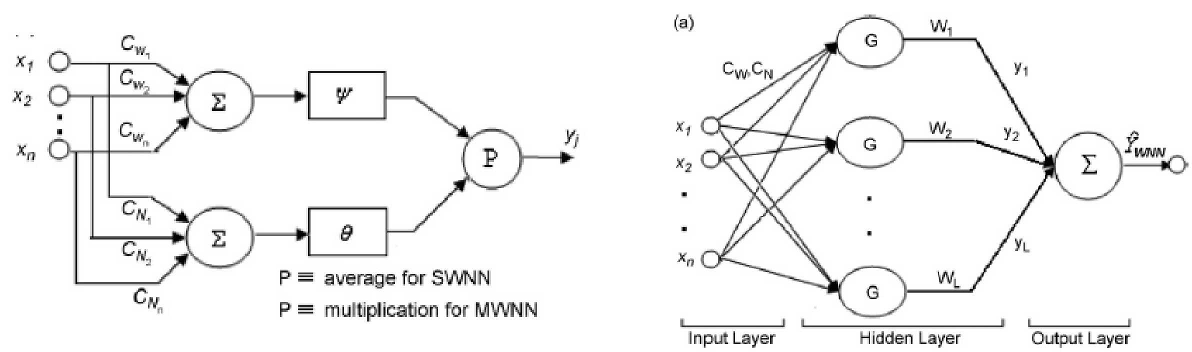


Fig **A2.2**: wavelet-sigmoid neuron

Neuron with two activation functions

Extreme leaning machine learns SLF_NNs very efficiently. To circumvent the limitations, SW-ELM is proposed [30]. Instead of a single TF in a hidden neuron,

ELM

- Initiation of parameters
- Complexity of architecture
- Activation function choice
- Remedy: SW_ELM

SW-ELM NN

+ Better non-line transformation
+ Deals with low and high frequency signals simultaneously
+ Small number of hidden neurons (i.e. compact structure

conjunction operation of two distinct activation functions (TF1, TF2) is used in each hidden neuron.

Appendix A3.a: Recurrent neurons (Rec_PEs)

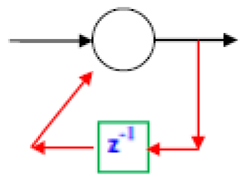
Neuron	Connection tensor	Features
	1 1⌈1	▶ Single neuron ▶ feedback connection

Fig. A3.1: A neuron with self feedback

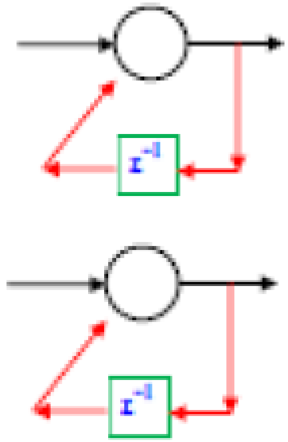
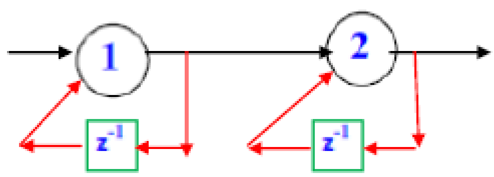
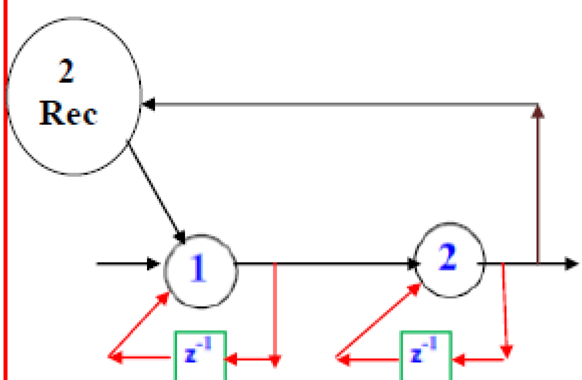
Two Neuron(s)	Connection	
	Tensor	Features
	$\begin{array}{c\|cc} & 1 & 2 \\ \hline 1 & 1 & 0 \\ 2 & 0 & 1 \end{array}$	▶ Not connected ○ Self feed-back
	$\begin{array}{c\|cc} & 1 & 2 \\ \hline 1 & 1 & 1 \\ 2 & 0 & 1 \end{array}$	▶ Connections → Feed-forward ← Self feed-back ← No feed-back
	$\begin{array}{c\|cc} & 1 & 2 \\ \hline 1 & 1 & 1 \\ 2 & 1 & 1 \end{array}$	▶ Connections → Feed-forward → Self feed-back → feed-back output of 2 to input 1

Fig. A3. 2: Two neurons with self-feedback connections

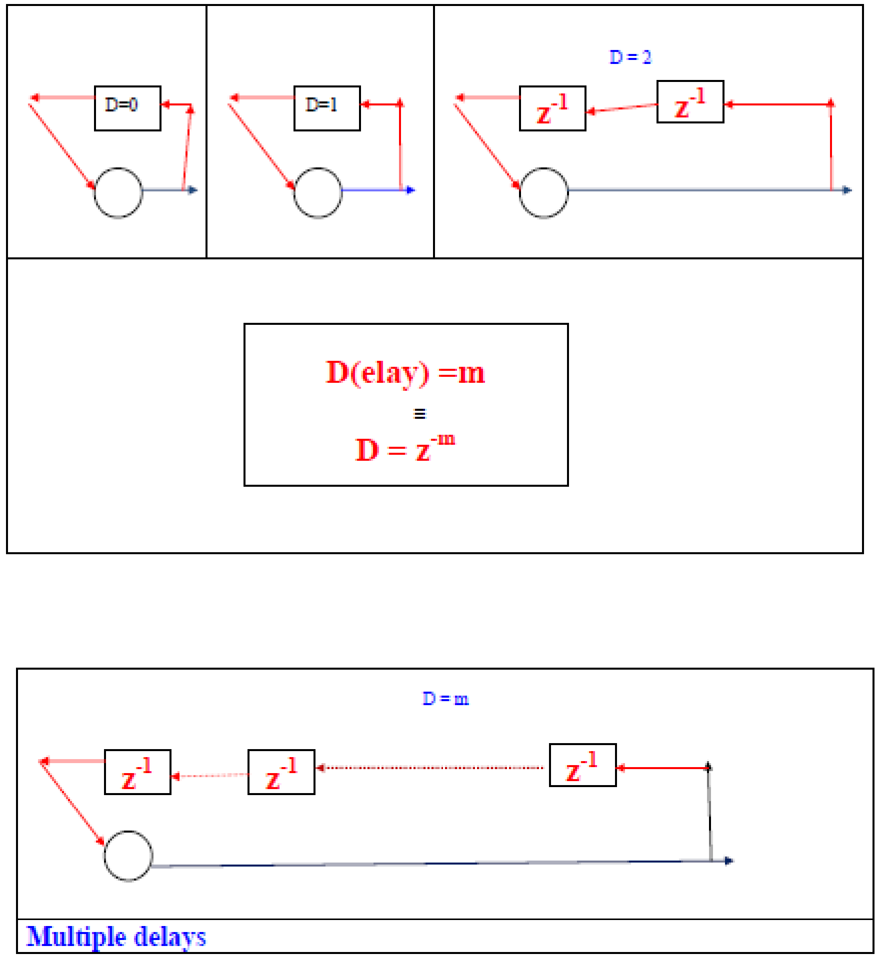
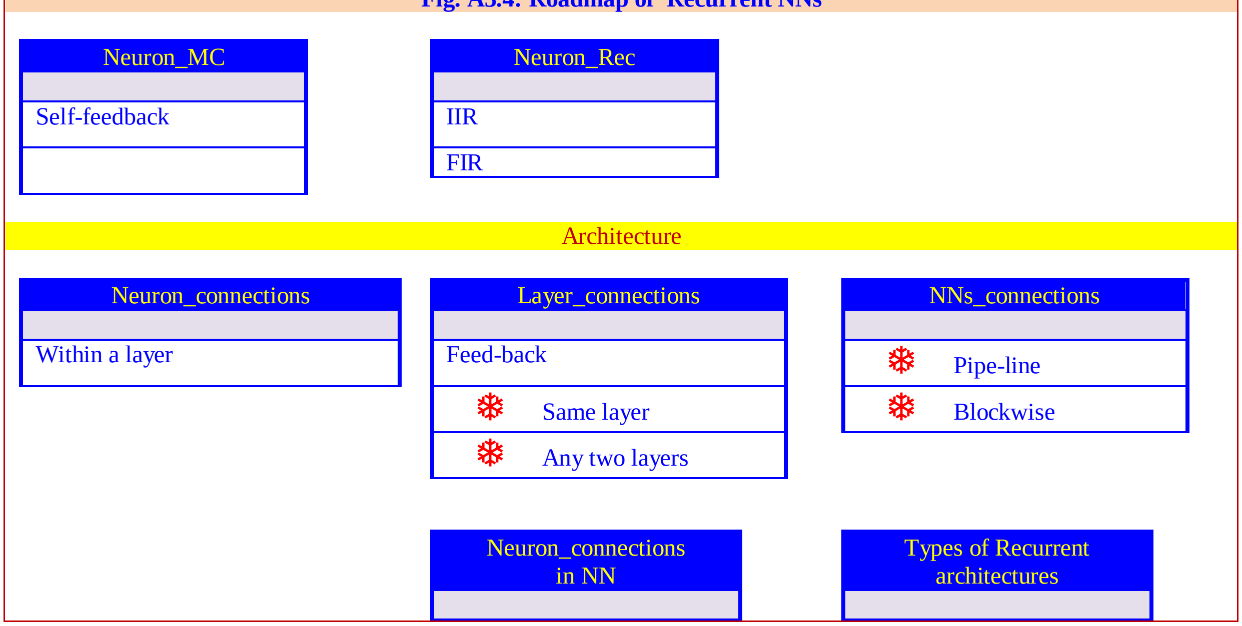
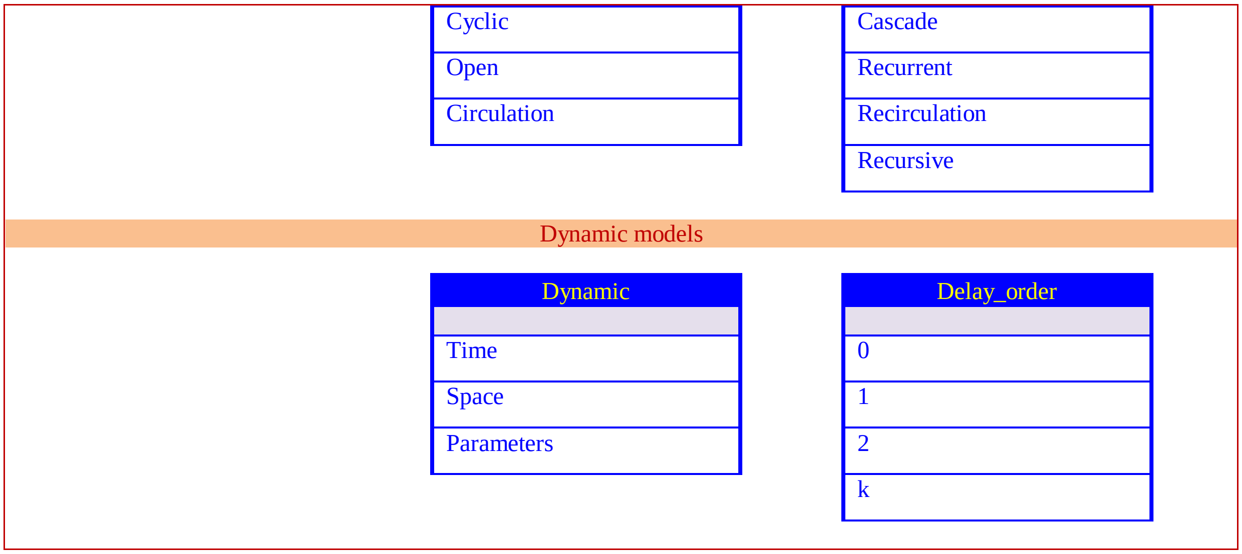


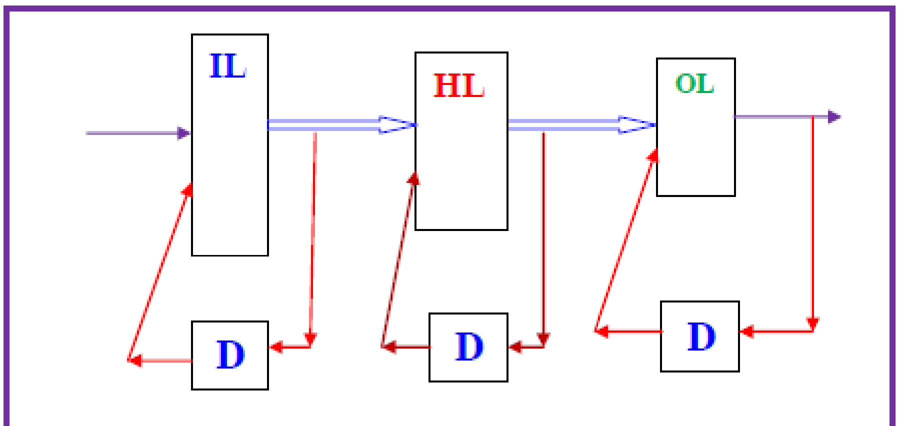
Fig. A3.3: Delays of multiple order in TS data

Fig. A3.4: Roadmap of Recurrent NNs





Appendix A3.c: Recurrent Multilayer Perceptron (Rec_MLP)



Self feed in layers of SLP ; D(elay) = 0;

Appendix A3.d: Time series-Rec_NNs

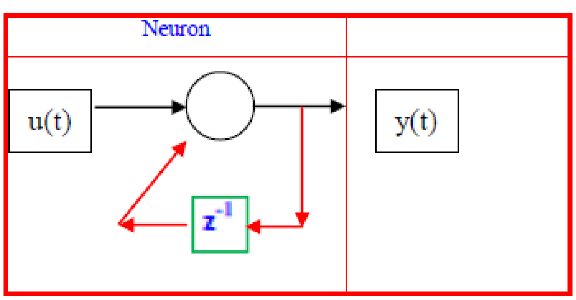


Fig. A3.5: Recurrent neuron in time series

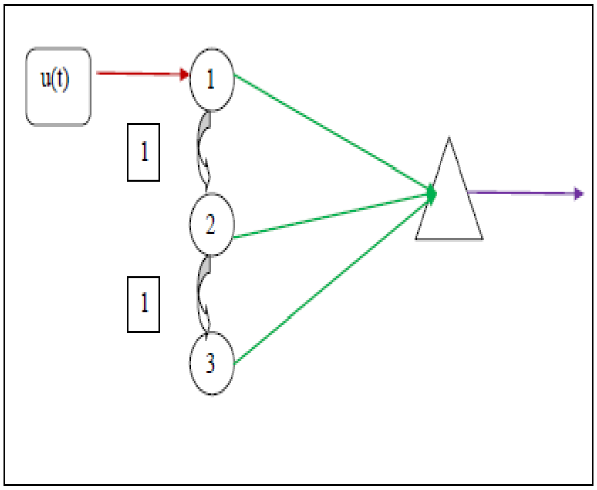


Fig. A3.6: I/O-NN model of time series and nonlinear-DEs

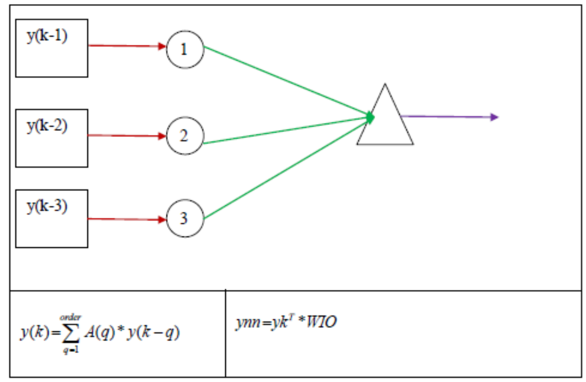


Fig. A3.7: I/O-NN model of sequence (time series) data. k= sequence; t: time series

$y(t)=TF_{(t)}\left(u(t)*w(t)+\left[TF_{(t-1)}\{u(t-1)*w(t-1)\}\right]\right)$		
If	Delay_order = 1 & W = 'purelin'	$y(t)=\left(u(t)*a(t)+\left[\{u(t-1)*\mathrm{a}(t-1)\}\right]\right)$
If	Delay_order = 0 & TF(t) ='SG' TF(t-1) ='purelin'	$y(t)=Sig\left(u(t)*w(t)+\left[\{u(t-1)*w(t-1)\}\right]\right)$
If	Delay_order = 0 & TF(t) ='Sig' TF(t-1) ='RBF'	$y(t)=Sig\left(u(t)*w(t)+\left[RBF\{u(t-1)*w(t-1)\}\right]\right)$

KB. A3. 1: Time series for delay orders and activation (TF) functions

Appendix A4: State_Of_the_Knowledge_of_the- (SOK-) Time_Series_Models (TSM)

The response of a mega to micro-processes with the lapse of time (on different scales viz. minutes to years) found a niche in the annals of data as 'Time Series'. Monitoring single/multiple response(s) for a sample as a function of time is called generally time series data. In pure chemistry, based on number of species whose concentrations change during the course of reaction (femato seconds to thousands of years), the discipline grew as chemical kinetics of fast reactions to radio-active decay. Here, the rate of progress of reaction is a non-linear function of concentrations of species and the constants of the model bear chemical significance reflecting how fast or how slow the reaction proceeds. Time series prediction can be a very useful tool in the field of process_chemometrics to forecast and to study the behaviour of key_process_parameters_in_time. Industrial processes, monitoring schedules where molecular level and reaction type are not known, it is still considered as time series. In all other disciplines, meteorology, physics, statistical data analysis of TS is coveted.

Objective of TS models: The primary concern is to develop relationship between the current observations with the previous ones in the time domain. The goal or sub-goal is to forecast value at (one-, two-, multi-step ahead) time instants in the future. The subtle interest of governmental agencies in time series for the retrospective inspection and development of control/ eradication measures for environmental pollution and consequent ill effects on human health is a seriously persuaded task of top priority. The term 'now cast' is also referred to as forecast, for instance ozone or pollution level within an hour/eight-hour period. The variations in time or space are similar in mathematical sense.

Data Structure_TS: Input values are only lagged real values of response (chart A4.1) and do not require explanatory variables in like in regression analysis for cause effect models. But, the restriction is that data should be stationary.

Chart A4.1: Data structure of time series

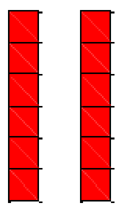
(a) Univariate time series

Univariate TS

$$time : \left[t_1 \quad t_2 \quad t_i \quad t_{np} \right]^T$$

$$u \quad : \left[u_1 \quad u_2 \quad u_i \quad u_{np} \right]^T$$

vectors



Abbreviations

NP	:	Number of (data) points
t_i	:	Time at which ith response is measured
u_i	:	i^{th} response at t_i
x_i	:	i^{th} exogeneous variable at t_i

If Data is vector & Auto-correlated
Then Time series

Univariate TS with eXogeneous variables

Univariate TS

eXogenerous variable

$$x \quad : \left[x_1 \quad x_2 \quad x_i \quad x_{np} \right]^T$$

orthogonal/ correlated data

If Data is matrix & correlation
Then Orhogonalisation methods

If Data is matrix & Correlation =0
Then Variables are not correlated

If Data is a matrix & Vector angles = 90
Then Variables are independent

The current response sometimes depends upon m-previous values and the data set is then auto-correlated. When, there is no trend after finite differences of auto-correlated function (ACF), the time series is stationary. But, in many real life problems non-stationary time series prevail. The input values are auto correlated and this makes the accurate estimation of individual response coefficients difficult. A way out to decrease the impact of collinearity is to increase sample size. But, the pragmatic issues are

Chart A4.2: Negative features of TS data

- Lack of large training data - Shortage of degrees of freedom - Severe multi collinearity data	- Dos not adhere to identity (stationarity) assumption - Process is not in statistical equilibrium

- Local optima, - Over fitting - Dimension disasters	- Multi-variate joint distribution of process change with time or space

Multiple TS data: As the complexity of the task grows, multiple time series arise. The consequence is TS data is high dimensional and is not easily reducible to two- three-dimensions.

Classical TS models

Abbreviations		Full form
MA	:	Moving Average
AR	:	Auto regression
ARMA	:	Autoregressive Moving Average
NARMA	:	Non-linear ARMA
ARIMA	:	AR Integrated MA
NAR MAX	:	Non-linear ARMA with eXogenous input

In time series models, time itself is considered as independent (pseudo_explanatory) variable unlike in regression and optimization procedures. The methods used to forecast non-linear real-world time series are broadly categorized as stochastic/ fuzzy set, NNs and nature-inspired (Eman) modules.

The simplest, without a ever possible competing candidate, is persistent model advocating the same behavior in the next time step. It means that response with respect to time is constant, but the observed perturbation (deviation) is due the then assumed concept of normal errors. It gave way to zero order (polynomial) model (with parameters --mean and standard deviation--) to detect outliers of physical significance. The dependence of current observation on past data in time led to moving average model (MA). Classical time series analysis considers (linear) trend, seasonal variation and spikes as components susceptible for estimation. The modeling is performed in a sequential manner after removing the spikes. The sum of responses of trend-model and season-model must account for the observed response. In this case, the residuals between observed and calculated values are only a random (or white) noise. The data is made stationary by detrending, calculating first and second order differences and Fourier transformation.

Linear-TS models

AR, ARMA, ARIMA are the linear time-series models accounting for the effect of mean, first order processes with different lags, while ARMAX takes into consideration of exo-geneous variables. An expert system based approach consists of 43 rules, with relative weight. A few typical rules (Alg. A4.1) attempt to combine or identify mixed model when both AR and MA models are plausible.

Alg. A4.1: Algorithm

Step : 1 Calculate ACF and PACF coefficients
Step : 2 select error criterion
Step : 3 If 95% of samples fall within the error criterion
Then White noise &
no analysis
Step : 4 Match shape of the ACF and of the PACF with set for simulated models

$\exp(-\beta * x)$

Step : 5 If Series is AR(1) type
Then PACF exhibits a spike

% Estimate F and y

Model	Coefficients
AR(1)	F1
AR(2)	$[F1, F2]^T$
MA(1)	[y1]
MA(2)	$[y1 , y2]^T$
ARMA(1,1)	[F1 , y1]

$Y_t = c1^* Y_{t-1} + c2^* y_{t-2} + e$

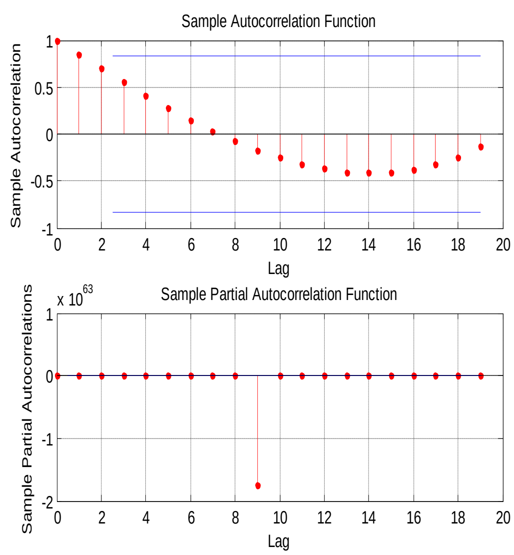
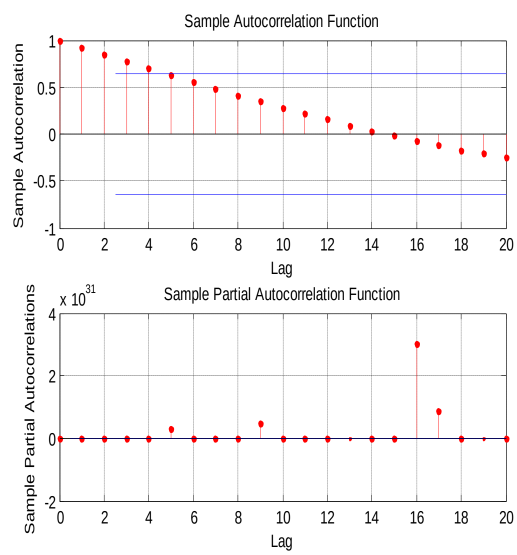
$$c1 + c2 < 1$$
$$c1 - c2 < 1$$
$$-1 < c2 < 1$$

Match the coefficients with those for experimental data

Step : 6 # ACF and PACF coefficients > (70% * error criterion)
Step : 7 Stationary condition
Step : 8 Find changes of sign in the ACF and the PACF
% Best model fitted

If PACF (> 70% above the error criterion) falls more abruptly than the ACF
Then model is AR(p)

If ACF (> 70% above the error criterion) falls more abruptly than the PACF
Then model is MA(q)
q is the number of ACF

ACF	ACF
	
y = [1:20]'; omacf(y);	y = [1:40]';


```
function [Lags,ACF,Bounds]= omacf(y)
[ACF,Lags, Bounds] = autocorr(y, [], 2);
[Lags,ACF],y',
ACF'
figure,subplot(211),autocorr(y, [], 2); subplot(212), parcorr(y)
%
% Matlab m file : autocorr.m
```

ACF	:	Auto correlation function
PCF	:	Partial correlation function

ARIMA
It is used in forecasting financial market with fluctuations also.

Periodic models: The periodicity is modeled with spectral analysis techniques like FT, peridiogram, and noise is sometimes filtered with Kalman filter and its advances.

Chart A4.3: ARIMA
— Axiom: Feature values of TS exhibits linear relationship with current and past values
— Data contain only white noise
— Requires a large amount of past data

Limitations of linear models: The time series data of real-world phenomenon like weather forecasting are rarely pure linear or nonlinear, but often contains both. Thus, failure of many parametric and non-parametric methods in vogue extensively for short and long term time series forecasting is usual than exception. The approximations of ARX series of models to complicated time series data are inadequate. The non-linearity and chaos further complicates the structure of profile. Recent methods viz, VAR, TAR, VEC and GARCH tackle complicated time series trends.

Abbreviations		Full form
VAR	:	Vector AR
ARCH	:	Autoregressive Hetero skedasticity
GARCH	:	Generalized ARCH
TAR	:	Threshold Autoregression
VEC	:	Vector Error Correction

Neural networks

The application of NNs in time series (TS) analysis is multifold. In an elementary jargon, the residual time series after accounting for ARMA type models is analyzed with black box FF-NNs. This approach is a hybrid model retaining the transparency of the classical models, and at the same time developing empirical model for non-linear part. The implicit assumption here is the validity of algebraic additivity of lagged-linear and non-linear components parts of time evolving phenomena.

In the next stage, MLP and SLP have been used with lagged inputs to forecast the future trend non-linear time series data. In FF-NNs, the inputs are responses with varying delays (i.e. lags: 1 to delays). SLP s use MC-neuron with confluence operator of summation of product of weights and input. The activation (or transfer) function (like sigmoid, RBF) produce non-linear mapping of weighted sum of inputs. In MLP-NNs with more than one HL, the output is multiplication of weighted sum of inputs.

Recurrent NNs had a niche in the complicated TS task. The general Rec-NN model for space or time with varying delays is noteworthy. Each node in Rec-NN has a self-feedback loop (connecting itself), in addition to connections to all other nodes. In other words RNNs have closed paths in the topology in certain architectures. Thus Rec_NNs model spacio-temporal data as it preserves the past states, while it is farther than FF-NNs' scope. The advances in NN and the imbibing character paved way for dedicated NNs implementing all types of linear and non-linear stochastic models viz. ARMA, ARIMA, ARMAX, NARMA, NARMAX etc. In time series analysis, data from discrete time series, differential equation (Mackey-Glass/inverted pendulum), difference equations (plant operation), full second order algebraic equations with equal/unequal constraints are analysed by Rec_NNs with remarkable results over classical methods and FF-NNs. For the bench mark data sets viz. sunspot data, exchange rate as well as other real life tasks like sand bar in beach, two-input-two-output problem, the end results are superior.

Rec-NN is a single window system for [232] sequential tasks like speech recognition, adaptive control and generation/analysis of sequences. Here, the current entity depends on those in the earlier ones. In dynamic time series data, responses in the previous time instances have their effects in the next time step and in Rec_NN, the value of either output of hidden or output layer is available in processing the next pattern. It ranges form 1-time delay to finite (FIR) or infinite (IIR) number. The memories at all instances of the model of course have to be stored in a separate stack or file. The results of one-and multi-step-ahead prediction of TS datasets with the models TSK-FIS, AR, Rec-NN are compared.

ARIMA + NN + FIS

The exchange rate (USD to IRAN – RIALS) and gold price (gram/USD) forecast are successfully predicted.

ARIMA + NN + FIS
+ Overcomes linear limitation.

NARX-NN

NARX-NN is a state space model for nonlinear dynamic systems. The results on chaotic-laser TS and a variable bit rate (VBR) video traffic TS with NARX – TS outperform TD_NN and Elman NNs.

KB. A4.1:	NARX_NN
If	NARX-NN with one hidden layer & Linear TF
Then	NARX-NN reduces to Linear-ARX-NN
If	NARX-NN & time series data
Then	NARX-NN → FF-time-delay-NN

- The predictive accuracy is reduced.
- **Remedy:** NARX-NN can be applied to a long term (multi-step-ahead) prediction of univariate TS.

Butterfly effect: The perturbations of results of model due to initialization procedures with random numbers are called butterfly effect. The consequence is long term predictions are prone to be in error.

Present status of TS

During the last one-decade, NN-models are compared mostly with MLR (ARMA)and rarely with non-linear models. Instead, comparison amongst different NNs (Rec_NN, hybrid_Rec_NNs, MLP, RBF, Fuzzy-ARTMAP, Kohonen,spiking_NNs etc) with advanced training algorithms will have a cool welcome to uphold the superiority of data driven paradigms over yester years model driven techniques. The human expert/analyst does not get a priori information and thus a hybrid system outperforms/ superior to the component models even in prediction. Rec-NNs on the other hand, excelled many of the hither to available non-linear time series models.

Still, like any other field, the choice of a method is a herculean task in TS research. It goes by tradition, expert advice, availability of expert system based software and cost in terms of human resource and time. Another hurdle is whether to use a most promising individual method or hybrid/hierarchical/sequential set of procedures. Recent trend is to analyze with a large number of methods of varying complexity, statistical measures for validation and considering best set of models and not thinking of only the unique best model. Spacio temporal NNs tackle simple as well as complex responses resulted in variation of time and/or space. The chaotic time series is applied in forecast of weather, finance and predictions of power load, hydrological data and sun spot profiles using ensemble methods, multi- dimension prediction with Lyapunov exponents, Rec_NN, NARX, wavelet neural networks etc. One of the trend setting packages is 'PREDICT' from Neural Ware Corporation. A pool of forecasting methods, feature sets and meta-learning techniques for time series data has been identified. At this juncture, experts intelligently use distribution/ information/ fuzzy characteristics/ wavelet/ ridgelet/ chaotic behavior of data/ noise and methods from classical/advanced statistics, NNs and nature-(including bio-) inspired procedures in prediction The prime objective of whole framework is to gain knowledge regarding best method for a task on hand. Instead of random combination of methods, ranking based approach is superior to single model selection. It is augmented by ensemble approach and Pareto optimal strategy for multiple_objective_goals.

Future scope of TimeSeries (FST) → Knowledge Intelligence Data structure (KIDs)

Future ventures with FFNNs, Rec.NNs using the state-of-the-art algorithms and encapsulating the modules in sequential and hierarchical mode will start a new era in handling TS data retaining the desired characteristics and eliminating limitations of the component procedures. Simulated data sets of multi-response TS with exogenous factors planned with statistical experimental design, S/N ratio and information content of various complications in trend, seasonality, spikes, chaos and noise structure result in feasibility study of mega NN structures. The ensemble study offers robustness for perturbation.

The usage 'state-of-the-art-' started in eighteenth century with an understanding that the word 'art' includes skills, methods, arts relating to the manufacturing and craftsmanship and not in the true sense of performing arts and fine arts. Around 1985, it is felt that it is over used to an extent that it has no punch left and sounds light if not like a lie as marketing strategy or appraisal of a tool. Recently, a new form 'state-of-the-$$$', 'state-of-knowledge-of-$$$' instead of 'state-of-the-art-of-$$$' is in vogue. Here knowledge again comprises of art, science, technology of (natural) universe including those of man-made ventures.

REFERENCES

[1] K RamaKrishna, V. Anantha Ramam, R. Sambasiva Rao, *J. Applicable Chem.*, **2014**, 3 (3), 919-989.

[2] K RamaKrishna, V. Anantha Ramam, R. Sambasiva Rao, *J. Applicable Chem.*, **2014**, 3 (2), 834-884.

[3] K. RamaKrishna, R. Sambasiva Rao, *J. Applicable Chem.*, **2014**, 3 (2), 449-492.

[4] K RamaKrishna, Ch. V. Kameswara Rao, V. Anantha Ramam, R. Sambasiva Rao, *J. Applicable Chem.*, **2014**, 3 (1), 006-029 (Online)

[5] K RamaKrishna, G.Ramkumar and R Sambasiva Rao , *J. Applicable Chem.*, **2013**, 2 (6), 1413-1458 (Online)

[6] K RamaKrishna, Ch V Kameswara Rao and R Sambasiva Rao, *J. Applicable Chem.*, **2013**, 2 (5), 1007-1034 (Online)

[7] K RamaKrishna, Ch V Kameswara Rao, R Sambasiva Rao, *J. Applicable Chem.*, **2013**, 2 (4): 698-713 (Online)

[8] M.V. Subaba Rao, V. AnantaRamam, V. Muralidhara Rao, R. Sambasiva Rao, *Research J. Science and Tech.*, **2013**, 5(4): 412-420

[9] K. Viswanath, R. Sambasiva Rao, Ch V. Kameswara Rao, K. Rama Krishna, B. Rama Krishna, G E G. Santhosh, *J. Applicable Chemistry*, **2012**,1, 109-124

[10] M.V. Subba Rao, V. AnantaRamam, V. Muralidhara Rao, R. Sambasiva Rao, *Asian J Chem*, **2010**, 22, 5937-5950.

[11] I. Suryanarayana, A. Braibanti, R. Sambasiva Rao, V. AnantaRamam, D. Sudarsan, G. Nageswara Rao, *Fisheries Research*, **2008**, 92, 115-139.

[12] R. Sambasiva Rao, G. Nageswara Rao, *Himalya Publisher*, **2005**, New Delhi (India).

[13] A Braibanti, R.Sambasiva Rao, V Ananta Ramam, G Nageswara Rao, and V V Panakala Rao, *Ann Chim.*, **2005**, 95(5), 291-301.

[14] L.H.A Rehman, K Sambasiva Rao, G. Nageswara Rao, R. Sambasiva Rao, *J. Ind. Council. Chemists,* **2001,** 17, 33-50.

[15] R Sambasiva Rao, *DRDE project report, Gwalior,* **2001.**

[16] G.Nageswara Rao, K.Sambasiva Rao, R.Sambasiva Rao and LH. Abdel Rehman, ***J.*** *Indian Chem.Chem*, **2000**, 17, 48

[17] K. Sambasiva Rao, R. Sambasiva Rao, *J. Korean Phys. Soc.*, **1998**, 32, S1850-S1851.

[18] A. Braibanti, K Sambasiva Rao, Ch S V Prasad and R. Sambasiva Rao, *Samyak J Chem.*, **1997**, 1(1),17-21.

[19] R. Sambasiva Rao, *Chem.Sci.* **1993**, 14(1), 23-44.

[20] K.V. Bapanaiah, R. Sambasiva Rao, *Science Reporter*, **1977**, 12, 315.

[21] Grossberg, *In R. Rosen and F. Snell (Eds., Progress in theoretical biology.* **1974,** *New York: Academic Press.*

[22] S. Grossberg, *Studies in Applied Math.*, **1973,** 52, 217-257.

[23] S. Grossberg, *Mathematical Bioscience,* **1972,** 15, 253-28.

[24] S. Grossberg, *Mathematical Bioscience,* **1972,** 15, 39.

Elman Rec_NN

[25] F. Fourati, *NeuroComput.* **2014, (**In Press).

[26] J. Guo, L. Ding, H. Luo, C. Zhou, L. Ma, *Tunnelling and Underground Space Technology,* **2014,** 41, 137-151.

[27] M. Malleswaran, V. Vaidehi, N. Sivasankari,*Aerospace Sci. Tech.*,32,1,**2014,**169-179.
[28] C. Hsu, *NeuroComput.* **2014,** 136, 170-179.
[29] A. Zolfagharian, A. Noshadi, M. R. Khosravani, M. Z. Md. Zain, *Appl. Math. Mod.,* **2014,** 38, 9–10, 2435-2453.
[30] K. Javed, R. Gouriveau, N. Zerhouni, *NeuroComput.* **2014,** 123, 299-307.
[31] A. Deihimi, A. Solat, I*nt. J. Elec. Power Energy Syst.*, **2014,** 54, 408-424.
[32] P. Li, Y. Li, Q. Xiong, Y. Chai, Y. Zhang, I*nt. J. Elec. Power Energy Syst.*, **2014,** 55, 749-759.
[33] M.G. De Giorgi, D. Bello, A. Ficarella, *Measurement,* **2014,** *47, 971*-981.
[34] L. Jinyang, M. Xiaofeng, *Eng. Appl. Art. Intel.*, **2013,** 26,4,1237-1245.
[35] C. Lin, I*nt. J. Elec. Power Energy Syst.*, **2013,** 52, 143-160.
[36] S. Mahmoud, A. Lotfi, C. Langensiepen, *Appl. Soft Comput.*, **2013,** 13,4, 1813-1822.
[37] C. Zhou, L.Y. Ding, R. He, *Automat. Const.*, **2013,** 36, 208-217.
[38] J. Zhao, X. Zhu, W. Wang, Y.Liu, N*euro Comput.*, **2013,** 118, 215-224.
[39] C. Hong, Ting-Chia Ou, Kai-Hung Lu, *Energy,* **2013,** *50, 270*-279.
[40] R. Köker, *Inf. Sci.*, **2013,** 222, 528-543.
[41] A. S. Al-Araji, M. F. Abbod, H. S. Al-Raweshidy, *NeuroComput.* **2013,** 99, 543-554.
[42] A. Zolfagharian, A. Noshadi, M. Z. Md. Zain, A. R. A. Bakar, *Swarm and Evolutionary Computation,* **2013,** 8, 54-68.
[43] X. Wang, L. Ma, B. Wang, T. Wang, *Neuro Comput.*, **2013,** 120, 547-559.
[44] X. Chen, C. Shen, *Optik – Intel. J. Light and Electron Optics*, **2013,** 124, 9, 784-792.
[45] L. Xu, S. Liu, *Mathematical and Comput. Modelling,* **2013,** 58, 3–4, 807-813.
[46] Y. Maali, A. Al-Jumaily, *Procedia Comput. Science,* **2013,** 24, 97-102.
[47] S. Kelo, S. Dudul, I*nt. J. Elec. Power Energy Syst.*, **2012,** 43, 1, 1063-1071.
[48] R. Chandra, M. Zhang, *Neuro Comput.*, **2012,** 86, 116-123.
[49] K. Siwek, S. Osowski, *Eng. Appl. Art. Intel.* , **2012,** 25, 6, 1246-1258.
[50] V. Nourani, M. S. Fard, A*dv. Eng. Software*, **2012,** 47, 1, 127-146.
[51] L. Wang, Y. Zhang, C. Yin, H. Zhang, C. Wang,*Simulation Modelling Practice and Theory,* **2012,** 25, 148-162.
[52] W. Cheng, J. Huang, C. Liou, *Knowledge-Based Syst.*, **2012,** 26, 271-280.
[53] Y. Tian, J. Zhang, J. Morris, *Ind. Eng. Chem. Res.*,**2001**, 40 (21), 4525–4535.
[54] A. Marvuglia, A. Messineo, *Energy Procedia,* **2012,** 14, 45-55.
[55] D.A. Asfani, A.K. Muhammad, Syafaruddin, M.H. Purnomo, T. Hiyama,*Expert Syst. Appl.*, **2012,** 39,5, 5367-5375.
[56] J. Petrović, S. Ibrić, G. Betz, Z. Durić, I*nt. J. Pharmaceutics*, **2012,** 428,1–2, 57-67.
[57] R.F. Hamade, A.H. Ammouri, H. Artail, *Eng. Appl. Art. Intel.*, **2012,** 25, 3, 628-639.
[58] I. Z. M. Darus, Ali A. M. Al-Khafaji, *Eng. Appl. Art. Intel.*, **2012,** 25, 1, 94-106.
[59] A. Vučković, F. Sepulveda, *Medical Eng. Physics,* **2012,** 34, 7, 964-971.
[60] K. Gobakis, D. Kolokotsa, A. Synnefa, M. Saliari, K. Giannopoulou, M. Santamouris, *Sustainable Cities and Society,* **2011,** 1, 2, 104-115.
[61] D. Niu, F. Wang, L. Zhang, D. He, M. Jia, *Chemomet. Intel. Lab. Syst.*, **2011,** 105, 1, 125-130.
[62] X. Wu, Q. Huang, X. Zhu, J. *Power Sources*, **2011,** 196, 3, 1295-1302.
[63] S.I. Ao, V. Palade, A*ppl. Soft Comput.*, **2011,** 11,2, 1718-1726.
[64] C. Wang, Z. Zhong, R. Li, E. Jia-qiang ,*Powder Tech.*, **2011,** 205,1–3, 201-207.
[65] C. Chupong, B. Plangklang, *Energy Procedia,* **2011,** 9, 230-237.

[66] W. Wan, H. Xu, W. Zhang, X. Hu, G. Deng, N*euro Comput.*, **2011,** 74,17, 2834-2841.

[67] N. Y. Steiner, D. Hissel, Ph. Moçotéguy, D. Candusso, I*nt. J. Hydrogen Energy,* **2011,** 36,4, 3067-3075.

[68] I. A. Gheyas, L. S. Smith, *Neuro Comput.,* **2011,** 74,18, 3855-3864.

[69] W. Heidl, C. Eitzinger, M. Gyimesi, F. Breitenecker, *Math. Comput. Simul.,* **2011,** 82, 3, 442-449.

[70] S. R. Mohanty, V. Ravikumar Pandi, B.K. Panigrahi, N. Kishor, P. K. Ray, A*ppl. Soft Comput.,* **2011,** 11, 8, 4789-4797.

[71] H. Zhou, G. Su, G. Li, *Procedia Comput. Science,* **2011,** 5, 754-758.

[72] D. West, S. Dellana, I*nt. J. Forecasting,* **2011,** 27, 3, 777-803.

[73] M. U. Suleman, M. M. Awais, *Robotics and Comput.-Integrated Manufacturing,* **2011,** 27, 4, 794-801.

[74] A. Cui, H. Xu, P. Jia,*Expert Syst. Appl.*, **2011,** 38, 7, 8186-8192.

[75] K. Baddari, N. Djarfour, T. Aïfa, J. Ferahtia, *J. Petroleum Science and Eng.,* **2010,** 71, 3–4, 106-111.

[76] N. Yousfi Steiner, D. Candusso, D. Hissel, P. Moçoteguy, *Math. Comput. Simul.,* **2010,** 81, 2, 58-170.

[77] L. Gao, S. Ren, *Anal. Biochem.*, **2010,** 405, 2, 184-191.

[78] M. Bildirici, E. A. Alp, O. O. Ersin, *Expert Syst. Appl.,* **2010,** 37, 1**,** 2-11.

[79] S. Papavlasopoulos, M. Poulos, N. Korfiatis, G. Bokos, I*nf. Sci.,* **2010,** 180, 11**,** 2156-2175.

[80] G. Xiao, Z. Zhu, *Tribology International,* **2010,** 43, 1–2, 218-227.

[81] M. Ardalani-Farsa, S. Zolfaghari, *Neuro Comput.*, **2010,** 73, 13–15, 2540-2553.

[82] A. Bahar, C. Özgen, *Eng. Appl. Art. Intel.,* **2010,** 23, 2, 262-270.

[83] N. Subrahmanya, Y. C. Shin, *Neuro Comput.,* **2010,** 73, 13–15, 2624-2631.

[84] B. Fan, Z. Du, X. Jin, X. Yang, Y. Guo, *Build. Env.,* **2010,** 45, 12, 2698-2708.

[85] T. E. Portegys, *Neural Networks,* **2010,** 23, 2, 306-313.

[86] S. Pravin Kumar, N. Sriraam, P.G. Benakop, B.C. Jinaga, *Expert Syst. Appl.,* **2010,** 37, 4, 3284-3291.

[87] B. A. Gross, D. M. Hanna, *Pattern Recog.,* **2010,** 43, 11, 3910-3921.

[88] G. Dede, M. H. Sazli, *Digital Signal Process.,* **2010,** 20, 3, 763-768.

[89] J. Lobato, P. Cañizares, M. A. Rodrigo, C. G. Piuleac, S. Curteanu, J. J. Linares, I*nt. J. Hydrogen Energy,* **2010,** 35, 15, 7889-7897.

[90] A. Amrouche, M. Debyeche, A. Taleb-Ahmed, J. M. Rouvaen, M. C.E. Yagoub, *Eng. Appl. Art. Intel.,* **2010,** 23, 1, 85-94.

[91] E. Portillo, M. Marcos, I. Cabanes, A. Zubizarreta, *Eng. Appl. Art. Intel.,* **2009,** 22, 8, 1270-1283.

[92] C. H. Aladag, E. Egrioglu, C. Kadilar, A*ppl. Math. Let.,* **2009,** 22, 9, 1467-1470.

[93] B. Karaağaç, M. İnal, V. Deniz, *Materials Design,* **2009,** 30, 5, 1685-1690.

[94] X. Tong, Z. Wang, H. Yu, *Comput. Physics Commun.*, **2009,** 180, 10, 1795-1801.

[95] J. Lobato, P. Canizares, M. A. Rodrigo, J. J. Linares, C. Piuleac, S. Curteanu, *J. Power Sources,* **2009,** 192, 1, 190-194.

[96] N. Sriraam, A*EU – Inter. J. Electronics and Commun.,* **2009,** 63, 4, 311-320.

[97] E. D. Übeyli, *Digital Signal Process.,* **2009,** 19, 2, 320-329.

[98] S. Chen, F. Lin, K. Shyu, N*euro Comput.*, **2009,** 72, 13–15, 3220-3230.

[99] S. Ekici, S. Yildirim, M. Poyraz, A*ppl. Soft Comput.*, **2009,** 9, 1, 341-347.

[100] A. Moghaddamnia, R. Remesan, M. Hassanpour Kashani, M. Mohammadi, D. Han, J. Piri, *J. Atmos. Solar-Terrestrial Phys.*, **2009,** 71, 8–9, 975-982.

[101] E. Salazar-Ruiz, J.B. Ordieres, E.P. Vergara, S.F. Capuz-Rizo, *Env. Mod. Software,* **2008,** 23, 8, 1056-1069.

[102] H. Ge, F. Qian, Y. Liang, W. Du, L. Wang, *Nonlinear Analysis: Real World Appl.,* **2008,** 9, 4, 1345-1360.

[103] B. Cartling, *Neuro Comput.,* **2008,** 71, 7–9, 1527-1537.

[104] M. Beccali, M. Cellura, V. Lo Brano, A. Marvuglia, *Renewable and Sustainable Energy,* **2008,** 12, 8, 2040-2065.

[105] A. Kirschner, D. Frishman, *Gene,* **2008,** 422, 1–2, 22-29.

[106] P. Ciarlini, U. Maniscalco, *J. Comput. Appl. Math.,* **2008,** 221, 2, 302-309.

[107] U. Brunelli, V. Piazza, L. Pignato, F. Sorbello, S. Vitabile, *Build. Env.,* **2008,** 43, 3, 304-314.

[108] J. M. P. Menezes Jr., G. A. Barreto, N*euro Comput.,* **2008,** 71,16–18, 3335-3343.

[109] N. Djarfour, T. Aïfa, K. Baddari, A. Mihoubi, J. Ferahtia, *Comptes Rendus Geoscience,* **2008,** 340, 6, 335-344.

[110] F. Thiery, S. Grieu, A. Traoré, M. Barreau, M. Polit, *Math. Comput. Simul.,* **2008,** 76,5–6, 388-397.

[111] F. Xue, K. Ke, *Syst. Eng. - Theory Practice,* **2008,** 28, 1**,** 40-45.

[112] A. Mellit, S. A. Kalogirou, *Progress in Energy and Combustion Science,* **2008,** 34, 5, 574-632.

[113] H. Wang, Y. Yu, G. Wen, Neural netw., **2014**, 55, 98-109.

[114] G. Pallocchia, E. Amata, G. Consolini, M.F. Marcucci, I. Bertello, J. *Atmos. Solar-Terrestrial Phys.,* **2008,** 70, 2–4, 663-668.

[115] F. Fourati, M. Chtourou, *Simulation Modelling Practice and Theory,* **2007,** 15, 8, 1016-1028.

[116] H. Ge, Y. Liang, M. Marchese, *Comput.s & Structures,* **2007,** 85, 21–22, 1611-1622.

[117] J. C. Gutiérrez-Estrada, C. Silva, E. Yáñez, N. Rodríguez, I. Pulido-Calvo, *Fisheries Res.,* **2007,** 86, 2–3, 188-200.

[118] M. Čerňanský, M. Makula, L. Beňušková, *Neural Networks*, **2007,** 20, 2, 236-244.

[119] U. Brunelli, V. Piazza, L. Pignato, F. Sorbello, S. Vitabile, *Atm. Env.,* **2007,** 41, 14, 2967-2995.

[120] W. Zhang, C. Bai, G. Liu,*Ecol. Model.,* **2007,** 201, 3–4, 317-325.

[121] M. Pacella, Q. Semeraro, *Comput. & Industrial Eng.,* **2007,** 52, 4, 502-520.

[122] J. Liu, D. Djurdjanovic, J. Ni, N. Casoetto, J. Lee, *Comput. Ind.,* **2007,** 58, 6, 558-566.

[123] I. Maqsood, A. Abraham, *Appl. Soft Comput.,*7 ,3, **2007,** 995-1004.

[124] C. Lin, J. Huang, M. Gen, G. Tzeng, *Appl. Math. Comput.,* **2006,** 175, 2, 1139-1146.

[125] D. Benaouda, F. Murtagh, J.-L. Starck, O. Renaud, *Neuro Comput.,* **2006,** 70, 1–3, 139-154.

[126] İ. Güler, E. D. Übeyli, *Pattern Recog. Let.,* **2006,** 27,13, 1560-1571.

[127] R. Köker, I*nf. Sci.,* **2006,** 176, 12, 1781-1799.

[128] S. Ren, L. Gao, J. *Electroanalytical Chem.,* **2006,** 586, 1, 23-30.

[129] L. Darren Graham, Doug R. Forbes, Simon D. Smith, A*utomat. Const.,* **2006,** 15, 5, 656-663.

[130] I. Maqsood, M. R. Khan, G. H. Huang, R. Abdalla, *Eng. Appl. Art. Intel.* , **2005,** 18, 1, **2005,** 115-125.

[131] J. D. Martin, Yu T. Morton, Q. Zhou, *Adv. Space Res.,* **2005,** 36, 12, 2480-2485.

[132] W. Huang, Y. Nakamori, S. Wang,*Comput.s & Operations Research,* **2005,** 32, 10, 2513-2522.

[133] J. B. Mbede, P. Ele, C. Mveh-Abia, Y. Toure, V. Graefe, S. Ma, I*nf. Sci.,* **2005,** 171, 4, 447-474.

[134] N. F. Güler, E. D. Übeyli, İ. Güler, *Expert Syst. Appl.,* **2005,** 29, 3, 506-514.

[135] R. Köker, *Eng. Appl. Art. Intel.,* 18, 6, **2005,** 685-693.
[136] J. C. Valle-Lisboa, F. Reali, H. Anastasía, E. Mizraji, N*eural Networks,* **2005,** 18 ,7, 863-877.
[137] J. Shieh, C. Chou, S. Huang, M. Kao, *Neuro Comput.,* **2004,** 57, 239-256.
[138] F. Temurtas, R. Gunturkun, N. Yumusak, H. Temurtas,*Elec. Power Syst. Res.,* **2004,** 72, 1, 33-40.
[139] A. Kalinli, D. Karaboga, *Eng. Appl. Art. Intel.,* **2004,** 17, 5, 529-542.
[140] G.I. Sainz, M.J. Fuente, P. Vega, *Euro. J. Control,* **2004,** 10, 1, 84-96.
[141] Y. Al-Assaf, R. El-Khazali, W. Ahmad,*Chaos, Solitons & Fractals,* **2004,** 22, 4, 897-905.
[142] S.L. Ho, M. Xie, T.N. Goh, *Comput. Math., Appl.,* **2003,** 46, 7, 1037-1045.
[143] P.R Patnaik, *Biochem. Eng. J.,* **2003,** 15, 3, 165-175.
[144] C. Leung, L. Chan, N*eural Networks,* **2003,** 16, 2, 223-239.
[145] J. S. Magnuson, B. McMurray, M. K. Tanenhaus, R. N. Aslin, *Cognitive Science,* **2003,** 27, 2, 285-298.
[146] V. Cho, *Tourism Management,* **2003,** 24, 3, 323-330.
[147] U Thissen, R van Brakel, A.P de Weijer, W.J Melssen, L.M.C Buydens, *Chemomet. Intel. Lab. Syst.,* **2003,** 69, 1–2, 35-49.
[148] E. Ayaz, S. Şeker, B. Barutçu, E. Türkcan, *Progress in Nucl. Energy,* **2003,** 43, 1–4, 381-387.
[149] S. Şeker, E. Ayaz, E. Türkcan, *Eng. Appl. of Art. Intel.,* **2003,** 16, 7–8, 647-656.
[150] R. S. Toqeer, N.S. Bayindir, *NeuroComput. ,* **2003,** 55, 3–4, 727-730.
[151] S. Watanabe, E. Sagawa, K. Ohtaka, H. Shimazu, *Adv. Space Res.,* **2003,** 31, 4, 829-834.
[152] X.Z. Gao, S.J. Ovaska, A.V. Vasilakos, *Comput. Commun.,* **2002,** 25, 16, 1477-1486.
[153] M. M. Abdelhameed, F. A. Tolbah, *Mechatronics,* **2002,** 12, 4, 617-633.
[154] D.J. Stein, J. Ludik, *Medical Hypotheses,* **2000,** 55, 2, 168-176.
[155] A. Sfetsos, A.H. Coonick, *Solar Energy,* **2000,** 68, 2, 169-178.
[156] D.T Pham, D Karaboga, A*rt. Intel. Eng.,* **1999,** 13, 2, 107-117.
[157] H. Wang, X. Yan, *Ind. Eng. Chem. Res.,* **2014**, 53 (16), 6789–6797.
[158] P.R Patnaik, *Biochem. Eng. J.,* **1999,** 3, 2, 113-120.
[159] R. Maksimovic, M. Popovic, *Medical Eng. Physics,* **1999,** 21, 5, 313-327.
[160] D. L.T Rohde, D. C Plaut, *Cognition,* **1999,** 72, 1, 67-109.
[161] T. Tiira, *Comput.s & Geosciences,* **1999,** 25, 8, 929-938.
[162] K. Brudzewski, *Sensors and Actuators B: Chem.,* **1998,** 47, 1–3, 231-234.
[163] A. C. Tsoi, A. Back, *Neuro Comput.,* **1997,** 15, 3–4, 183-223.
[164] P.R. Patnaik, *Process Bio chem.,* **1997,** 32, 391-400.
[165] A. Sperduti, N*eural Networks,* **1997,** 10, 3, 395-400.
[166] H. Lundstedt, *Physics and Chem. Earth,* **1997,** 22, 7–8, 623-628.
[167] W. DeLiang, L. Xiaomei, S. C. Ahalt, *Neural Networks,* **1996,** 9, 7, 1099-1118.
[168] T. Ziemke, *Pattern Recog. Let.,* **1996,** 17, 4, 319-334.
[169] B. H. Jansen, P. R. Desai, I*nt. J. Bio-Medical Comput.,* **1994,** 37,3, 249-257.
[170] D.T. Pham, X. Liu, A*rt. Intel. Eng.,* **1993,** 8, 1, 67-75.
[171] K. Shimohara, T. Uchiyama, Y. Tokunaga, *Neural Networks,* **1993,** 6, 5, 709-718.
[172] P.R. Patnaik, *Chem. Eng. J.,* 84, 3, 533-541.

Jordan Rec_NN

[173] H. Wang, G. Song, *Neuro Comput.,* **2014,** 134, 289-295.
[174] E. C. Carcano, P. Bartolini, M. Muselli, L. Piroddi, *J. Hydrology,* **2008,** 362, 3–4, 291-307.

[175] M. Matouq, T. El-Hasan, H. Al-Bilbisi, M. Abdelhadi, M. Hindiyeh, S. Eslamian, S. Duheisat, *J. Taibah University for Science*, **2013,** 7, 2, 44-55.
[176] H. Abu Qdai32s, K. Bani Hani, N. Shatnawi, *Resources, Conservation and Recycling*, **2010,** 54, 6, 359-363.
[177] H. A. Bekhet, S. F. K. Eletter, *Review of Development Finance*, **2014 (press).**
[178] E.D. Mehleri, P.L. Zervas, H. Sarimveis, J.A. Palyvos, N.C. Markatos, *Renewable Energy*, **2010,** 35, 7, 1357-1362.
[179] E.C. Gonçalves, L.A. Minim, J.S.R. Coimbra, V.P.R. Minim, *Chem. Eng. Process.*, **2005,** 44, 12, 1269-1276.
[180] W. A. Truccolo, G. Rangarajan, Y. Chen, M. Ding, *Neural Networks*, **2003,** 16, 10, 1453-1460.
[181] S. Darsono, J. W. Labadie, *Env. Mod. Software*, **2007,** 22, 9, 1349-1361.
[182] M. Ishikawa, T. Moriyama, *Fuzzy Sets and Syst.*, 82, 2, **1996,** 167-176.
[183] S. MacNamara, P. Cunningham, J. Byrne, *Inf. Process. & Manage.*, **1998,** 34, 4, 395-403.
[184] J. Wu, L. Jiao, *Physica D: Nonlinear Phenomena*, **2008,** 237, 19, 2487-2498.
[185] J. Wu, L. Jiao, *Physica A: Statistical Mech. Appl.*, **2007,** 386, 1**,** 469-480.
[186] J. Wu, L. Jiao, *Physica A: Statistical Mech. Appl.*, **2007,** 386, 1**,** 513-530.
[187] K. Chen, L. Xu, H. Chi, *Neural Networks*, **1999,** 12, 9**,** 1229-1252.
[188] E. Avci, *Expert Syst. Appl.*, **2013,** 40, 10, 3984-3993.
[189] E. Avci, *Expert Syst. Appl.*, **2007,** 32, 2, 485-498.
[190] J. V. Hansen, *Inf. Sci.*, **1999,** 119, 1–2, 91-105.
[191] C. Yuan, M. J. Druzdzel, *Math. Comput. Model.*, **2006,** 43, 9–10, 1189-1207.
[192] M. Kobayashi, G. Dupret, O. King, H. Samukawa, *Comput. Math.Appl.*, **2001,** 42, 10–11, 1331-1352.
[193] A. Al-Anbuky, S. Bataineh, S. Al-Aqtash, *Control Eng. Practice*, **1995,** 3, 9, 1291-1298.

Recurrent_Neural Networks (Rec_NN)

[194] Q. Wu, T. M. McGinnity, L. Maguire, A. Belatreche, B. Glacki, N*eural netw.*, **2008,** 21, 9, 1318-1327.
[195] Y. Lv, W. Lv, J. Sun, N*onlinear Analysis: Real World Applications*, **2008,** 9, 4, 1590-1606.
[196] C Liou, J. Huang, W. Yang, *Neuro Comput.*, **2008,** 71, 16-18, 3150-3157.
[197] D. Calabuig, J.F. Monserrat, D. Gómez-Barquero, O. Lázaro, *Neuro Comput.*, **2008,** 71, 16-18, 3439-3446.
[198] C. Hsu, K. Cheng, *Neuro Comput.*, **2008,** 71, 16-18, 3447-3459.
[199] Y. Liu, Z. You, *Neuro Comput.*, 71, 16-18, **2008,** 3595-3601.
[200] M. Yeh, C. Wen, S. Chian, *NeuroComput.*, 71, 16-18, **2008,** 3685-3689.
[201] C. Xu, X. Tang, M. Lia, N*euroComput.*, 74, 5, **2011,** 689-707.
[202] J. A.K.Suykens, B. L.R.DeMoor, J. Vandewalle, N*eural netw.*, 7(5), **1994,** 819-831.
[203] F. Pasemann, *Neural netw.*, 7(3), 1**995,** 421-429.
[204] R. K.Brouwer, *Neural netw.*, 8(4), 1**995,** 597-603.
[205] C. W.Omlin, C.L. Giles, *Neural netw.*, 9(1), 1**996,** 41-52.
[206] M. Tajine, D. Elizondo, N*eural netw.*,11, 1**998, 1**571-1588.
[207] M. Kimura, R. Nakano, N*eural netw.*, 11, 1**998, 1**589-1599.
[208] T. Lin, B. G. Horne, C.L. Giles, *Neural Networks*, 11, 5, 1**998,** 861-868.
[209] D. P.Mandic, J. A.Chambers, *Neural Comput.*, 11, **1999, 1**069-1077.
[210] D.P. Mandic, J.A. Chambers, *Neural netw.*, *1*2, **1999, 1**341–1345.

[211] K. W. C. Ku, S. M. W. Mak, W. C. Siu, *IEEE Trans. Neur. Netw., 1*0(2), 1**999,** 239-252.
[212] E. Gelenbe, J. Fourneau, *Neural Comput., 1*1, 1**999,**953-963.
[213] M. C. M. Elliffe, E. T.Rolls, N.Parga, A. Renart, *Neural netw.,* 2**000,** *1*3, 225-237.
[214] F. A.Gers, J. Schmidhuber, F. Cummins, *Neural Comput.,* 2**000,** *1*2, 2451-2471.
[215] Y. Xia, J. Wang, *Neural netw.,* 2**000,** *1*3, 337-350.
[216] K. Arai, R. Nakano, N*eural Networks,* 2**000,** *13,* 6, 667-680.
[217] E. Trentin, D. Giuliani, *Neural Comput. Applic.,* 2**001,** *1*0, **1**20–135.
[218] M. Benson, R. A. Carrasco, *Neural Comput. Applic.,* 2**001,** *1*0, **1**36–147.
[219] A. G.Parlos, S. K. Menon, A. F.Atiya, I*EEE Trans. Neur. Netw.,* 2**001,** 12(6), 1411-1432.
[220] W.H.Goh, C.P.Lim, K.K.Peh, K.Subari, N*eural Comput. Applic.,* 2**001,** 10**,** 311-317.
[221] A. Blanco, M. Delgado, M.C. Pegalajar, N*eural Networks,* 2**001,** 14, 1, 93-105.
[222] A.Blanco, M.Delgado, M.C.Pegalajar, N*eural netw.,* 2**001,** 14, 93-105.
[223] I. Gabrijel, A. Dobnikar, N*eural Networks,* 2**003,** *16,* 1,101-120.
[224] S. L. Goh, D. P. Mandic, N*eural Networks,* 2**003,** 16, 8, 1095-1100.
[225] J. A. Perez-Ortiz, F. A. Gers, D. Eck, J. Schmidhuber, N*eural netw.,* 2**003,** 16, 241-250.
[226] S. K. Chalup, A. D. Blair, N*eural Networks,* 2**003,** 16, 7**,** 955-972.
[227] Y. Xia, G. Fenga, J. Wangb, N*eural netw.,* 2**004,** 17, 1003–1015.
[228] Z. Xua, H. Qiaob, J. Penga, B. Zhangc, N*eural netw.,* 2**004,** 17, 73–85.
[229] P. Baldi, M. Rosen-Zvic, N*eural netw.,* 2**005,** 18, 1080–1086.
[230] Sanqing Hu, Derong Liu, N*eural Networks,* 2**005,** 18, 2, 171-178.
[231] E. Leclercq, F. Druaux, D. Lefebvre,S. Zerkaoui, N*eurocomput.,* 2**005,** 63, 25–44.
[232] U. D. Schiller, J. J. Steil_, N*euro comput.,* **2005,** 63**,** 5–23.
[233] A. Micheli, F. Porter, A. Sperduti, N*euro comput.,* **2005,** 64, 73–92.
[234] T. Ren, T. Chen, *Electric Power Systems Research,* **2006,** 76, 12, 1064-1074.
[235] H. Ohta, Y. P. Gunjib, N*eural netw.,* 2**006,** 19, 1106–1119.
[236] Z. Zeng, J. Wang, N*eural Networks,* 2**006,** *19,* 10, 1528-1537.
[237] İ. Güler, E. D. Übeyli, *Pattern Recognition Letters,* 2**006,** *2*7, 13, 1560-1571.
[238] J.B.Theocharis, *Fuzzy sets Syst.,* 2**006,** 157, 471-500.
[239] I. Szita, A. Lőrincz, N*eurocomputing,* 2**006,** *70,* 1–3, 577-591.
[240] Y. Liu, Z. Wang, X. Liu, N*eural Networks,* 2**006,** 19, 5, 667-675.
[241] C. Lin, C. Chen, *European Journal of Operational Research,* 2**006,** *172, 2,* 696-715.
[242] S. L. Goh, D. P. Mandic, N*eural Networks,* 2**007,** *2*0, 10, 2**007,** 1061-1066.
[243] B. Chen, J. Wang, N*eural Networks,* 2**007,** *20,* 10, 1067-1080.
[244] B. Chen, J. Wang, N*eural netw.,* 2**007,** *2*0(10), 1067-1080.
[245] Y. Sun, J. Cao, N*onlinear Analysis: Real World Applications,* 2**007,** 8, 4, 1171-1185.
[246] C. Juang, T. Chen, N*eurocomputing,* 2**007,** 71, 1–3, 121-130.
[247] T. D. Ganchev, D. K. Tasoulis, M. N. Vrahatis, N. D. Fakotakis, N*eurocomputing, 70,* 7–9, 2**007,** 1424-1438.
[248] T.G. Barbounis, J.B. Theocharis, N*eurocomputing,* 70, 9, 2**007,** 1525-1542.
[249] T. Pan, R. Wang, J. Lai, A*dvances in Water Resources,* 30, 8, 2**007,** 1797-1814.
[250] J. Liang, J. Cao, N*onlinear Analysis: Real World Applications, 8,* 1, 2**007,** 187-197.
[251] X. Cai, N. Zhang, G. K. Venayagamoorthy, D. C. Wunsch , N*eurocomputing,*70,13–15, 2**007,** **2**342-2353.
[252] H. Williams, J. Noble, *Biosyst.,* 87, 2–3, 2**007,** **2**52-259.

[253] S. J. Yoo, J. B. Park, Y. H. Choi, *Information Sciences*, **2007,** 177, 15, 3074-3098

[254] A. Savran, *IEEE Trans. Neur. Netw.*, **2007,** 18(2), 373-384.

[255] Y. Liu, Z. You, L, Cao, *Comput. & Math. Applic.*, **2007,** 53, 1, 41-53.

[256] T. Li, S. Fei, *Neurocomputing*, **2007,** 71, 1–3, 428-438.

[257] Z. Hou, M. M. Gupta, P. N. Nikiforuk, M. Tan, L. Cheng, *IEEE Trans. Neur. Netw.*, **2007,** 18(2), 466-481.

[258] M. A. Wiering, I. L. Turner, *Neural netw.*, **2007,** *2*0, 509–518.

[259] T.G. Barbounis, J.B. Theocharis, *Inform. Sci.*, **2007,** 177, 24, 5775-5797.

[260] X. Cai, D. V. Prokhorov, D. C. Wunsch, *IEEE Trans. Neur. Netw.*, **2007,** 18(3), 674-684.

[261] I. N. da Silva, W. C. do Amaral, L. V. de Arruda, *Applied Math. Model.*, **2007,** 31, 1, 78-92.

[262] G. Peng, L. Huang, *Nonlinear Analysis: Hybrid Systems*, **2008,** *2*, 4, 1198-1204.

[263] R. T. Bambang, *Applied Soft Comput.*, **2008,** 8, 4, 1498-1504.

[264] R.K. Al Seyab, Yi Cao, *Comp. & Chem. Eng.*, **2008,** 32, 7, 1533-1545.

[265] X. Liao, Q. Luo, Z. Zeng, Y. Guo, *Nonlinear Analysis: Real World Applications*, **2008,** 9, 4, 1535-1557.

[266] J. Xu, Y. Cao, D. Pi, Y. Sun, *Neuro comput.*, **2008,** 71,7–9, 1566-1577.

[267] A. M. Schaefer, S. Udluft, H. Zimmermann, *Neurocomputing*, **2008,** 71, 13–15, **2**481-2488.

[268] Q. Song, *Neurocomputing*, **2008,** 71, 13–15, **2**823-2830.

[269] K. Jeong, D. Kim, J. Jung, M. Kim, G. Joo, *Ecolog. Model.*, **2008,** *211*, 3–4, 292-300.

[270] R. Setiono, B. Baesens, C. Mues, *IEEE Trans. Neur. Netw.*, **2008,** 19(2), 299-307.

[271] H. Zhang, Z. Yi, L. Zhang, *Comput. & Math. with Applic.*, **2008,** 56, 12, 3130-3137.

[272] J. Juang, H. Chiou, L. Chien, *Neurocomputing*, **2008,** 71, 16–18, 3224-3238.

[273] W. Yilei, Song Qing, and Liu Sheng, *IEEE Trans. Neur. Netw.*, **2008,** 19(2), 351-375.

[274] L. Liu, H. Shao, D. Nan, *Neurocomputing*, **2008,** 71, 16–18, 3589-3594.

[275] K. Nouri, R. Dhaouadi, N. B. Braiek, *Applied Soft Comput.*, **2008,** *8*, 1, 371-382.

[276] M. Assaad, R. Boné, H. Cardot, *Information Fusion*, **2008,** 9, 1, 41-55.

[277] R. Rakkiyappan, P. Balasubramaniam, *App. Math. Comput.*, **2008,** 198, 2, 526-533

[278] Q. Liu, J. Wang, *IEEE Trans. Neur. Netw.*, **2008,** 19(4), 558-570.

[279] R.K. Al Seyab, Y. Cao, *Journal of Process Control*, **2008,** *18*, 6, 568-581.

[280] K. Patan, *Neural netw.*, **2008,** 21, 59–64.

[281] W. Lee, K. Yang, *Neurocomputing*, **2008,** 71, 4–6, 600-610.

[282] D. Park, *Applied Mathematics and Computation*, **2008,** 205, 2, 648-657.

[283] P. Balasubramaniam, R. Rakkiyappan, *Applied Math. Comput.*, 204, 2, **2008,** 680-686.

[284] B. Gao, W. Zhang, *IEEE Trans. Neur. Netw.*,19(5), **2008,** 782-794.

[285] X. Xie, Z. Cao, X. Weng, D. Jin, *NeuroComput.*, 72, 04-06, **2009,** 1042-1055.

[286] Y. Zhang, D. Yue, E. Tian, *Neuro Comput.*, **2009,** 72, 4-6, 1265-1273.

[287] E. D. Übeyli, *Digital Signal Processing*, **2009,** *19*, 1, 134-143.

[288] E. Kolman, M. Margaliot, *Fuzzy Sets and Systems*, **2009,** 160, 2, 145-161.

[289] H. Zhao, J. Zhang, *Neural Networks*, **2009,** *22*, 10, 1471-1483.

[290] D. Graves, W. Pedrycz, *Neuro comput.*, **2009,** 72, 7–9, 1668-1678.

[291] C. Qiao, Z. Xu, *Neuro comput.*, **2009,** 72, 7–9, 1878-1886.

[292] H. Li, B. Chen, C. Lin, Q. Zhou, *Neuro Comput.*, **2009,** 72,7-9, 2017-2023.

[293] Z. Wang, Y. Liu, X. Liu, *Neural Networks*, **2009,** 22, 41-48

[294] R.A. Aliev, B.G. Guirimov, Bijan Fazlollahi, R.R. Aliev, *Fuzzy Sets and Systems, 160,* **2009,** 17, 2553-2566.
[295] C. Song, H. Gao, W. X. Zheng, *Neurocomputing,* **2009,** 72, 10–12, 2563-2568.
[296] X. Zhu, Y. Wang, G. Yang, *Neurocomputing,* **2009,** 72, 13–15, 3376-3383.
[297] B. Du, James Lam, *Neural Networks,* **2009,** *22,* 4, 343-347.
[298] W. Zhou, J. M. Zurada, *Neurocomputing,* 72, 16–18, **2009,** 3782-3788.
[299] L. Zhang, Z. Yi, J. Yu, P. A. Heng, *Neurocomputing,* **2009,** 72,16–18, 3809-3817.
[300] L. Li, L. Huang, A*pplied Mathematical Modelling,* **2009,** 33, 12, 4326-4336.
[301] E. Zio, M. Broggi, N. Pedroni, *Progress in Nuclear Energy,* **2009,** *51,* 3, 573-581.
[302] H. Xiang, J. Cao, N*onlinear Analysis: Theory, Methods & Applications,* **2009,** 71, 12, 6097-6108.
[303] C. Juang, C. Lai, C. Tu, *Expert Systems with Applications,* 2**009,** 36,3, 6368-6374.
[304] A. Carnell, N*euro Comput.,* **2009,** 72, 4-6, 685-692.
[305] S. Zerkaoui, F. Druaux, E. Leclercq, D. Lefebvre, *Engineering Applic. Art. Intel.,* **2009,** *2*2, 4–5, 702-717.
[306] C. Hsu, *Control Eng. Practice,* **2009,** *17,* 6, 714-722.
[307] J. Wang, Y. Hsu, H. Lin, Y. Chen, *Pattern Recognition Letters,* **2009,** 30, 9, 812-819.
[308] Y. Hsu, J. Wang, J*ournal of Process Control,* **2009,** 19, 6, 942-953.
[309] X. Zhao, N*onlinear Analysis: Theory, Methods & Applications,* **2009,** 71, 12, e2873-e2878.
[310] T. Ogata, S. Nishide, H. Kozima, K. Komatani, H. G. Okuno, *Pattern Recognition Letters, 31,* 12, **2010,** 1560-1569.
[311] P. Zheng, W. Tang, J. Zhang, N*euroComput.,*73,10-12, **2010,** 2280-2285
[312] S. Babaei, A. Geranmayeh, S. A. Seyyedsalehi, *Computer Methods and Programs in Biomedicine,* **2010,** 100, 3, 237-247.
[313] I. Sutskever, G. Hinton, N*eural Networks,* **2010,** *23,* 2, 239-243.
[314] H. Hu, H. Jiang, Z. Teng, N*euro Comput.,* **2010,** 73, 13-15, 2589-2596.
[315] M. Übeylï, E. D. Übeylï, *Expert Systems with Applications,* **2010,** 37, 4, 2742-2746.
[316] C. Feng, C. O'Reilly, R. Plamondon, *Neurocomputing,* **2010,** *74, 1*–3, 274-283.
[317] C. Qiao, Z. Xu, N*eurocomputing,* **2010,** *73,* 13–15, 2783-2788.
[318] F. Maleki, B. Beheshti, A. Hajihosseini, G. R. R. Lamooki, N*eurocomputing,* **2010,** *73,* 16–18, 3066-3078.
[319] J. Wang, Y. Hsu, *Neurocomputing,* **2010,** *74, 1*–3, 315-327.
[320] M. Hermans, B. Schrauwen, *Neural Networks,* **2010,** 23, 3, 341-355.
[321] X. Wang, Y. Huang, N. Nguyen, *Neurocomputing,* **2010,** *74, 1*–3, 354-361.
[322] X. Li, X. Fu, *J. Comput. Appl. Math.,* **2010,** 234, 2, 407-417.
[323] X. Li, X. Fu, P. Balasubramaniam, R. Rakkiyappan, N*onlin. Analysis: Real World Applications,* **2010**, *11, 5, 4092*-4108.
[324] M.U. Akhmet, D. Aruğaslan, E. Yılmaz, *Neural Netw.,* **2010**, 23, 7, 805-811.
[325] J. Qiu, *NeuroComput.,* **2010,**73, 4-6, 820-826.
[326] M. Amiri, H. Davande, A. Sadeghian, S. Chartier, *NeuroComput.* **2010**, *23, 7, 892*-904.
[327] A. Hajihosseini, G. R. R. Lamooki, B. Beheshti, F. Maleki, N*eurocomput.,* **2010,** 73,4–6, 991-1005.
[328] Y. Liu, B. Liu, S. Ling, *Neurocomp.,* **2011**,74, 6, 1062-1068.
[329] H. Zhao, X. Zeng, J. Zhang, T. Li, Y. Liu, X. Wang, T. Li, N*eural Networks,* **2011,** *24,* 1, 12-18.
[330] H. Zhang, W. Wang, B. Xiao, A*ppl. Math. Model.,* **2011,** 35, 1, 123-129.

[331] S.M. Kelo, S.V. Dudul, *Expert Systems with Applic.,* **2011**, 38, 3, 1554-1564.
[332] Y. Li, K. Zhao, *Neurocomput.,* **2011,** *74, 10,* 1632-1637.
[333] A. Cherif, H. Cardot, R. Boné, N*eurocomput.,* **2011,** *74,* 11, 1936-1944.
[334] S. Freitag, W. Graf, M. Kaliske, J. U. Sickert, *Computers & Structures,* **2011,** *89,* 21–22, 1971-1981.
[335] M. Syed Ali, M. Marudai, *Math. Comput. Model.*, **2011,** *54*, *9*–10, 1979-1988.
[336] T. Hsieh, H. Hsiao, W. Yeh, A*pplied Soft Computing,* **2011,** 11, 2, 2510-2525.
[337] M.R. Banaei, A. Kami, *Energy Conversion and Management,* **2011,** 52, 7, 2629-2636.
[338] L. Dehyadegary, S. A. Seyyedsalehi, I. Nejadgholi, *Neurocomputing,* **2011,** 74, 17, 2716-2724.
[339] M. Li, D. Liu, *Procedia Engineering,* **2011**, 15, 282-287.
[340] A. Wu, Z. Zeng , X. Zhu, J. Zhang, N*eurocomputing,* **2011,** 74, 17, 3043-3050.
[341] W. Hinoshita, H. Arie, J. Tani, H. G. Okuno, T. Ogata, N*eural Networks*, **2011,** *24,* 4, 311-320.
[342] Q. Wu, J. Zhou, L. Xiang, N*eurocomputing,* **2011,** *74,* 7, 3204-3211.
[343] R. Chandra, M. Frean, M. Zhang, C. W. Omlin, N*eurocomputing,* **2011,** *74,* 17, 3223-3234.
[344] Q. Liu, J. Cao, N*eurocomputing,* **2011,** 74, 17, 3494-3501.
[345] H. Zhao, X. Zeng, J. Zhang, T. Li, Y. Liu, D. Ruan, I*nformation Sciences,* **2011,** *181*, 17, 3677-3692.
[346] Z. Wang, L. Huang, N*euroComput.,* **2011,** 74, 17, 3695-3699.
[347] E. C. Yen, *Commun. Nonlinear Sci. Num. Simul.,* **2011,** *16*, 1, 509-521.
[348] C. Peng, G. D. Magoulas, A*ppl. Math. Comput.*, **2011,** *217*, 12, 5421-5441.
[349] Q. Luo, Z. Zeng, X. Liao, N*eurocomputing,* **2011,** 74, 4, 638-645.
[350] C. Xu, X. Tang, M. Lia, N*euroComput.*, **2011,**74, 689–707.
[351] H. Huang, G. Feng, N*eurocomputing, 74,* 5, **2011,** 792-796.
[352] C. Juang, Y. Lin, R. Huang, *Fuzzy Sets and Systems*, **2011,** *179,* 1, 83-99.
[353] X. Meng, M. Tian, S. Hu, N*eurocomputing,* **2011,** *74,* 6, 949-953.
[354] Gang Bao, Zhigang Zeng, N*eurocomputing,* **2012,** 77, 1, 101-107.
[355] A. Wu, S. Wen, Z. Zeng, I*nformation Sciences,* **2012,** *183,* 1, 106-116.
[356] A. Wu, Z. Zeng, N*eural Networks,* **2012,** *36,* 1-10.
[357] S. Li, S. Chen, B. Liu, Y. Li, Y. Liang, N*eurocomputing,* **2012,** *91,* 1-10.
[358] A. Banakar, M. F. Azeem, A*pplied Soft Computing,* **2012,** 12, 3, 1187-1200.
[359] D. Xu, Hongmei Shao, Huisheng Zhang, N*eurocomputing,* **2012,** 93, 133-136.
[360] Q. Cao, B. T. Ewing, M. A. Thompson, *European Journal of Operational Research,* **2012,** 221, 1, 148-154.
[361] G. Zhang, Y. Shen, J. Sun, N*eurocomputing,* **2012,** *97,* 149-154.
[362] Y. Yang, W. Chang, T. Lia, N*euroComput.*, **2012,** 82,179-185
[363] C. Qiao, Z. Xu, N*eurocomputing,* **2012,** *77,* 1, 205-211.
[364] Y. Huang, H. Zhang, Z. Wang, N*eurocomputing,* **2012,** *91*, 21-28.
[365] S. Wen, Z. Zeng, T. Huang, N*eurocomputing,* **2012,** *97,* 233-240.
[366] J. Cabessa, A. E.P. Villa, *Theoretical Computer Science,* **2012,** 436, 23-34.
[367] C. Tu, C. Juang, *Expert Systems with Applications,* **2012,** 39, 3, 2479-2488.
[368] T. Mingming, Z. Jinliang, J. *Econom. Business,* **2012,** *64,* 4, 275-286.
[369] Y. Zhao, Q. Liu, *Mathematics and Computers in Simulation,* **2012,** *86,* 2-9.
[370] K. K. Sarma, A. Mitra, A*EU - International Journal of Electronics and Communications,* **2012,** *66*, 4, 322-331.

[371] S. Vineetha, C. Chandra Shekara Bhat, Sumam Mary Idicula, *Gene,* **2012,** *506,* 2, 408-416.
[372] A. Eddahech, O. Briat, N. Bertrand, J. Delétage, J. Vinassa, I*nt. J. Elect. Power & Energy Syst.*, **2012,** *42,* 1, 487-494.
[373] S. C. Duong, E. Uezato, H. Kinjo, T. Yamamoto, A*utomation in Construction,* **2012,** *23,* 55-63.
[374] M. Zhang, Z. Chu, *Ocean Engineering,* **2012,** 45, 56-62.
[375] S. Babaei, A. Geranmayeh, S. A. Seyyedsalehi, *Expert Systems with Applications,* **2012,** *39,* 6, 6263-6274.
[376] E. Artun, T. Ertekin, R. Watson, B. Miller, *Computers & Geosciences,* **2012,** 38,1, 68-79.
[377] D. Monner, J. A. Reggia, N*eural netw.*, **2012,** 25, 70–83.
[378] J. Santos, Á. Campo, N*eurocomputing,* **2012,** *86,* 86-96.
[379] Q. Liu, Z. Guo, J. Wang, N*eural Networks,* **2012,** *26,* 99-109.
[380] J. Sun, J. Chen, A*pplied Mathematics and Computation,* **2013,** *221*, 111-120.
[381] J. Xing, H. Jiang, C. Hu, N*eurocomputing,* **2013,** *113,* 122-129.
[382] H. Huang, T. Huang, X. Chen, C. Qian, N*eural Networks,* **2013,** *48,* 153-157.
[383] V. Sharma, D. Srinivasan, *Engineering Applications of Artificial Intelligence, 26,* 5–**2013,** 6, 1562-1574.
[384] Z. Guo, J. Wang, Z. Yan, N*eural Networks,* **2013,** *48,* 158-172.
[385] S. Zhu, Y. Shen, N*eural Networks,* **2013,** *38,* 17-22.
[386] Y. Bai, J. Chen, N*eurocomputing,* **2013,** *121,* 179-184.
[387] D. Park, I*nformation Sciences,* **2013,** *237*, 18-28.
[388] H. Ling, S. Samarasinghe, D. Kulasiri, *Biosystems,* **2013,** *114,*3, 191-205.
[389] S. Wen, G. Bao, Z. Zeng, Y. Chen, T. Huang, N*eural Networks,* **2013,** *48,* 195-203.
[390] C. O. Galán, J.R. Rodríguez Pérez, S. García Cortés, A. Bernardo Sánchez, *Math. Comput. Model.*, **2013,** *57,*7–8, 2016-2023.
[391] S. Zhu, Y. Shen, N*eurocomputing,* **2013,** *113,* 220-226.
[392] Bing Li, D. Xu, N*eurocomputing,* **2013,** *103,* 239-246.
[393] R. Coban, *Engineering Applications of Artificial Intelligence,* **2013,** *26*, 1, 241-250.
[394] Q. Liu, T. Huang, N*eurocomputing,* **2013,** *118,* 263-267.
[395] S. Freitag, W. Graf, M. Kaliske, *Computers & Structures,* **2013,** *124*, 29-37.
[396] P. Liu, I*SA Transactions,* **2013,** *52,* 1, 30-35.
[397] Y. Ding, W. Jiang, P. Yu, *Communications in Nonlinear Science and Numerical Simulation,* **2013,** *18,* 2**,** 351-372.
[398] S. Zhu, Y. Shen, N*eurocomputing,* **2013,** *101,* 370-374.
[399] S. Vineetha, C. Chandra Shekara Bhat, S. M. Idicula, *Gene,* **2013,** *515,* 2, 385-390.
[400] G. Zhang, Y. Shen, Q. Yin, J. Sun, I*nformation Sciences,* **2013,** *232,* 386-396.
[401] C. Yi, Y. Zhang, D. Guo, *Mathematics and Computers in Simulation,* **2013,** *92,* 40-52.
[402] Y. Wang, C. Zheng, E. Feng, N*eurocomputing,* **2013,** *119,* 454-461.
[403] H. Huang, T. Huang, X. Chen, N*eural Networks,* **2013,** *46,* 50-61.
[404] S. Qin, W. Bian, X. Xue, N*eurocomputing,* **2013,** *120***,** 655-662.
[405] P. Chen, L. Chang, F. Chang, J*ournal of Hydrology,* **2013,** *497,* 71-79.
[406] L. Hou, H. Zhu, S. Zhong, Y. Zhang, Y. Zeng, N*eurocomputing,* **2013,** 115, 72-80.
[407] A. Hosseini, J. Wang, S. M. Hosseini, N*eural Networks,* **2013,** *44,* 78-86.
[408] K. J. Boström, H. Wagner, M. Prieske, M. de Lussanet, *Human Movement Science,* **2013,** 32, 5, 880-898.

[409] E. Egrioglu, U. Yolcu, C. H. Aladag, E. Bas,*Procedia - Social and Behavioral Sciences,* **2014,** *109*,1094-1100.

[410] Zaccheu O. Olaofe, *Sustainable Energy Technologies and Assessments,* **2014,** *6,* 1-24.

[411] W. Luo, K. Zhong, S. Zhu, Y. Shen, N*eural Netw.*, **2014,** *53*, 127-133.

[412] J. Xiao, Z. Zeng, A. Wu, N*eurocomputing*, **2014,** *134*, 182-188.

[413] Y. Huang, H. Zhang, Z. Wang, A*pplied Mathematics and Computation*, **2014,** *229*, 187-200; IEEE Trans. Neural Networks And Learning Systems, **2014,** 25(7), 1229-1262.

[414] R. Coban, A*nnals of Nuclear Energy,* **2014,** *69*, 260-266.

[415] Seong-Ik Han, Jang-Myung Lee, I*SA Transactions*, **2014,** *53*, 1, 33-43.

[416] H. Zhang, Y. Huang, B. Wang, Z. Wang, N*eurocomputing,* **2014,** *136*, 337-344.

[417] G. Chen, O. Gaans, S. V. Lunel, A*pplied Mathematics Letters*, **2014,** *27*, 36-42.

[418] J. Li, J. Yang, X. Yuan, Z. Hu, N*eurocomputing,* **2014,** *131*, 388-396.

[419] X. Zhou, J. Tian, H. Ma, S. Zhong, N*eurocomputing*, **2014,** *129*, 401-408.

[420] X. Ruan, Y. Zhang, N*eurocomputing*, **2014,** 129, 421-427.

[421] G. Wu, Z. Zhu, *Fuzzy Sets and Systems*, **2014,** *237*, 47-62.

[422] E. Kilic, M. Dolen, H. Caliskan, A. B. Koku, T. Balkan, *Control Eng. Practice,* **2014,** *26*, 51-71.

[423] S. Jeong, Y. Park, R. Mallipeddi, J. Tani, M. Lee, N*eurocomputing*, **2014,** 129, 67-77.

[424] M. Fairbank, S. Li, X. Fu, E. Alonso, D. Wunsch, N*eural Networks,* **2014,** *49*, 74-86.

[425] S. Qin, D. Fan, P. Su, Q. Liu, *Commun. Nonlinear Sci. Numer. Simul.,* **2014,** *19*,4, 789-798.

[426] G. Li, Z. Yan, J. Wang, N*eural Networks*, **2014,** *50*, 79-89.

[427] W. Wang, L. Li, H. Peng, J. Xiao, Y. Yang, N*eural Netw.*, **2014,** *53*, 8-14.

Cancer research

[428] S. Deweerdt, *Nature*, **2014,** 509, May, S60_S61.

[429] T. Davoli, A. Wei Xu, K. E. Mengwasser, L. M. Sack, J. C. Yoon, P. J. Park, S. J. Elledge, *Cell* , **2013,** 55, Nov.7, 948–962.

HIV

[430] S. Kalomoiris, J. Lawson, R. X. Chen, G. Bauer, J. A. Nolta, J. S. Anderson, *Human Gene Therapy Methods,* **2012,** 23, Dec., 366–375.

[431] M. Okamoto, H. Chono, Y. Kawano, N. Saito, H. Tsuda, K. Inoue, I. Kato, J. Mineno, M. Baba,*Human Gene Therapy Methods,* **2013,** 24, April, 94–103.

[432] A. Stornaiuolo, B. M. Piovani, S. Bossi, E. Zucchelli, S. Corna, F. Salvatori, F. Mavilio,C. Bordignon, G. P. Rizzardi, C. Bovolenta, *Human Gene Therapy Methods,* **2013,** 24, August, 228–240.