ECOS: EVOLVING CONNECTIONIST SYSTEMS AND THE ECO LEARNING PARADIGM

Nikola Kasabov

Email: nkasabov@otago.ac.nz
Department of Information science, University of Otago
P.O.Box 56, Dunedin, New Zealand

ABSTRACT

The paper presents a framework called ECOS for Evolving COnnectionist Systems. ECOS evolve through incremental learning. They can accommodate any new input data, including new features, new classes, etc. New connections and new neurons are created during operation. The ECOS framework is used to develop a particular type of evolving neural networks - evolving fuzzy neural networks. A novel training method is introduced and called eco training. ECOS are four (to six) orders of magnitude faster than the multilayer perceptrons, or fuzzy neural networks, trained with the backpropagation algorithm. This is illustrated on benchmark problems, as well as on a real-time problem such as the task of voice recognition and person identification.

1. SEVEN MAJOR REQUIREMENTS TO THE FUTURE CONNECTIONIST SYSTEMS

Many developers and practitioners in the area of neural networks (NN) have enjoyed the power of the traditional NNs when solving AI problems. At the same time they have noticed several difficulties when NN are applied to real world problems, such as speech and image recognition, adaptive prediction, adaptive on-line control, intelligent agents. These tasks usually require flexible learning and dynamically adaptive connectionist systems (COS) that have 'open' structures and are able to process both data and knowledge. Seven major requirements (that are addressed in the ECOS framework) are listed below:

(1) A COS should be able to learn quickly from large amount of data therefore using fast training, e.g. 'one-pass' training.

(2) A COS should be able to adapt in a real time and in an on-line mode where new data is accommodated as it comes.

(3) A COS should have an 'open' structure where new features (relevant to the task) can be introduced at a later stage of the system's operation, e.g., the system creates 'on the fly' new inputs, outputs, connections and nodes.

(4) A COS should be able to accommodate in an incremental way everything that is, and that will become known about the problem, i.e. in a supervised (instructional), or in an unsupervised mode, using one modality or another, accommodating data, rules, text, image, etc.

(5) A COS should be able to learn and improve through active interaction with other COSs and with the environment in a multi-modular, hierarchical fashion.

(6) A COS should adequately represent space and time in their different scales, inner spatial representation, short- and long-term memory, age, forgetting, etc.

(7) A COS should be able to analyse itself in terms of behaviour, global error and success; to explain what it has learned and what it 'knows' about the problem it is trained to solve; to make decision about its own improvement.

Several NN theories, models and methods for adaptive learning and for dynamical modification of NN structures have been introduced so far: incremental learning [1]; lifelong learning; on-line learning [3]; growing NN [2]; pruning NN [10,8,4]; etc. A framework called ECOS (Evolving COnnectionist System) that addresses all the issues above is introduced in the paper along with a method of training called eco training. They are illustrated on bench mark data and real-world data (Iris classification and voice recognition respectively).

2. THE PRINCPLES OF ECOS

ECOS are systems that evolve in time trough interaction with the environment, i.e. an ECOS adjusts its structure with a reference to the environment. A block diagram of the ECOS framework is given in fig.1. ECOS are multi-modular, hierarchical, open systems. Their main parts are described below.

(1) Presentation part. It performs perception and filtering of input information. The number of inputs (features) can vary from examples to examples.

(2) Representation and memory part, where information (patterns) are stored. It is a multi-modular, evolving structure of NN modules organised in spatially distributed groups, e.g. NNs in one group can represent phonemes in a spoken language (one NN representing one class phoneme).
(3) Higher level decision part. It consists of several modules, each taking decision on a particular problem (e.g., word recognition, face identification). The modules receive a feedback from the environment and make decision about the functioning and the adaptation of ECOS.
(4) Action part. The action modules take the output from the decision modules and pass information to the environment.
(5) Self-analysis, and rule extraction modules. This part extracts compressed abstract information from the representation modules and from the decision modules in different forms of rules, abstract associations, etc.

ECOS are multi-level, multi-modular structures where many modules are connected with inter- and intra-connections. The evolving connectionist system does not have to have a 'clear' multi-layer structure. An ECOS is a modular 'open' structure evolving over time. Initially it is a mesh of nodes (neurons) with very little connections between them, pre-defined through *prior* knowledge or 'genetic' information. These connections mainly connect modules of the initial connectionist structure. An initial set of rules can be inserted in this structure. Gradually, through self-organisation, the system becomes more and more 'wired'. The network stores different patterns (exemplars) from the training examples. A node is created and designated to represent an individual example if it is significantly different from the previous ones (with a level of differentiation set through dynamically changing parameters. The functioning of the ECOS from fig.1 is based on the following general principles.
(1) Input patterns are presented one by one, in a pattern mode, having not necessarily the same input feature sets. After each input example is presented, the ECOS either associates this example with an already existing rule (case) node, or creates a new one. A NN module, or a neuron is created when needed at any time of the functioning of the whole system. After the presentation of each new input example the system is able to react properly on both new and old examples.
(2) The representation module evolves in two phases. In phase one input vector **x** is passed through the representation module and the case nodes become activated based on the similarity between the input vector and their input connection weights. If there is no node activated above a certain *sensitivity threshold (Sthr)* a new rule neuron (*rn*) is created and its input weights are set equal to the values of the input vector **x** and the output weights - to the desired output vector. In phase two, activation from either the winning case neuron ("one-out of-n" mode), or from all case neurons with activation above an *activation threshold (Athr)* ("many-of-on" mode) is passed to the next level of neurons. Evolving can be achieved in both supervised and unsupervised modes. In a supervised mode the final decision which class (e.g., phoneme) the current vector **x** belongs to, is made in the higher-level decision module that may activate an adaptation process. Then the connections of the representation nodes to the class output nodes, and to the input nodes are updated with the use of *learning rate coefficients lr1 and lr2,* correspondingly. If the class activated is not the desired one, then a new case node is created. The feedback from the higher level decision module goes also to the feature selection and filtering part. New features may be involved in the current adaptation and evolving phase. In an unsupervised mode a new case node is created if there is no existing case node or existing output node that are activated above *Sthr* and an *output threshold Othr* respectively. The parameters *Sthr, lr1, lr2, Errthr, Athr and Othr* can change dynamically during learning.
(3) Along with growing, an ECOS has a pruning procedure defined. It allows for removing neurons and their corresponding connections that are not actively involved in the functioning of the ECOS thus making space for new input patterns. Pruning is based on local information kept in the neurons. Each neuron in ECOS keeps a 'track' of its 'age', its average activation over the whole life span, the error it contributes to, and the density of the surrounding area of neurons. Pruning is performed through the fuzzy rule:
IF case node (j) is OLD, and average activation of (j) is LOW, and the density of the neighbouring area of neurons is HIGH or MODERATE, and the sum of the incoming or outgoing connection weights is LOW,
THEN the probability of pruning node (j) is HIGH.
(4) The case neurons are spatially organised and each neuron has its relative spatial dimensions in regards to the rest of the neurons based on their reaction to the input patterns. If a new neuron is created when the input vector **x** was presented, it is allocated close to the neuron which had the highest activation to the input vector **x**.
(5) There are two global modes of learning in ECOS:
(a) Active learning mode - learning is performed when a stimulus (input pattern) is presented and kept active.
(b) Eco training mode - learning is performed when there is no input pattern presented at the input of the ECOS. In this case the process of further elaboration of the connections in ECOS is done in a passive learning phase, when existing connections that store previously 'seen' input patterns are used as eco training examples. The connection weights that represent stored input patterns are now used as exemplar input patterns for training other modules in ECOS. This type

of learning with the use of 'echo' data is called here eco training. There are two types of eco training: (1) cascade eco training; (2) sleep eco training. In *cascade eco training* a new NN module is created when a new class data is presented. The module is trained on the positive examples of this class, plus the negative examples of the following different class data, and on the negative examples of previously stored patterns in previously created modules. In the sleep eco training, modules are created with positive examples only when data is presented. Then the modules are trained on the stored in the other modules patterns as negative examples.

(6) ECOS provide explanation information extracted from the structure of the NNs. Each case (rule) node is interpreted as an IF-THEN rule as it is in the FuNN fuzzy neural network [5].

(7) ECOS are biologically inspired. Some biological motivations for evolving systems are given in [9].

(8) The ECOS framework can be applied to different types of NNs , neurons, activation functions etc. One realisation that uses the ECOS framework is the evolving fuzzy neural networks EFuNNs and the EFuNN algorithm as given in [7]. The EFuNN realisation of ECOS has been used in the experiments below to illustrate the main principles of ECOS on the Iris data set (150 instances; 3 classes - setosa, versicolour and virginica; four attributes - sepal length, sepal width, petal length, petal width. Three EFuNNs are evolved.

Experiment1. **Incremental, one-pass learning.** Characteristics of the evolved system: "one-of-n" mode; no pruning; radial basis activation function is used in the case neurons; Sthr=0.75; Errthr= 0.1; lr=0; rn(setosa)=22; rn(versicolor)= 27; rn(virginica)=25. Overall classification: Setosa-50(100%); Versicolor-47(88%); Virginica-43 (86%).

Experiment 2. **A second learning pass.** A second pass on the evolved in experiment 1 EFuNNs is performed. SThr=0.8; Errthr=0.05; rn(setosa) = 22; rn(versicolor) =37; rn(virginica)=37. Overall correct classification: Setosa - 50(100%); Versicolor - 50 (100%); Virginica - 50 (100%).

Experiment 3. **Using positive examples only.** The three EFuNNs are evolved by using positive examples only. SThr=0.85; Errthr=0.05; rn(setosa) = 6; rn(versicolor) =16; rn(virginica)=20. Overall classification: Setosa - 50(100%); Versicolor - 48 (96%); Virginica - 46 (92%). This also results in high activation of the EFuNNs when similar data, but from different classes, are presented.

Experiment 4. **Cascade-eco learning**. SThr=0.8, Errthr=0.1; rn(setosa) = 19 (4 positive); rn(versicolor) =33 (9 positive) ; rn(virginica)=28 (14 positive). Overall classification: Setosa - 50(100%); Versicolor - 48 (96); Virginica - 46 (92%).

Experiment 5. **Sleep eco training**. SThr=0.9; Errthr=0.05; rn (setosa) = 6; rn (versicolor) =16; rn (virginica)=20. Overall classification: Setosa - 50(100%); Versicolor - 50 (100%); Virginica - 46 (92%). The results of the sleep eco-training are better than the results after training with positive data only, but the significant difference is that here the false positive activation is eliminated, similar to the case in experiments 1, 2 and 4.

3. EFuNNs FOR ON-LINE ADAPTIVE VOICE AND PERSON RECOGNITION

Here a small experiment on voice data collected from a CNN news movie is presented [6]. 2.5 msec voice data is used as reference data for each of four TV show presenters. Voice-type data taken from sections of 1.5 msec are used for testing. The voice data is transferred every 11.8 msec (with 50% overlap between two consecutive windows) into 26-element mel scale (MS) vectors. The 26- element MS vectors are averaged over a time frame of 125 msec thus producing 20 examples for training and 10 examples for testing for each person. The evolved EFuNNs require four to six order of magnitude less time for training per input vector than the reported in [6] experiments.

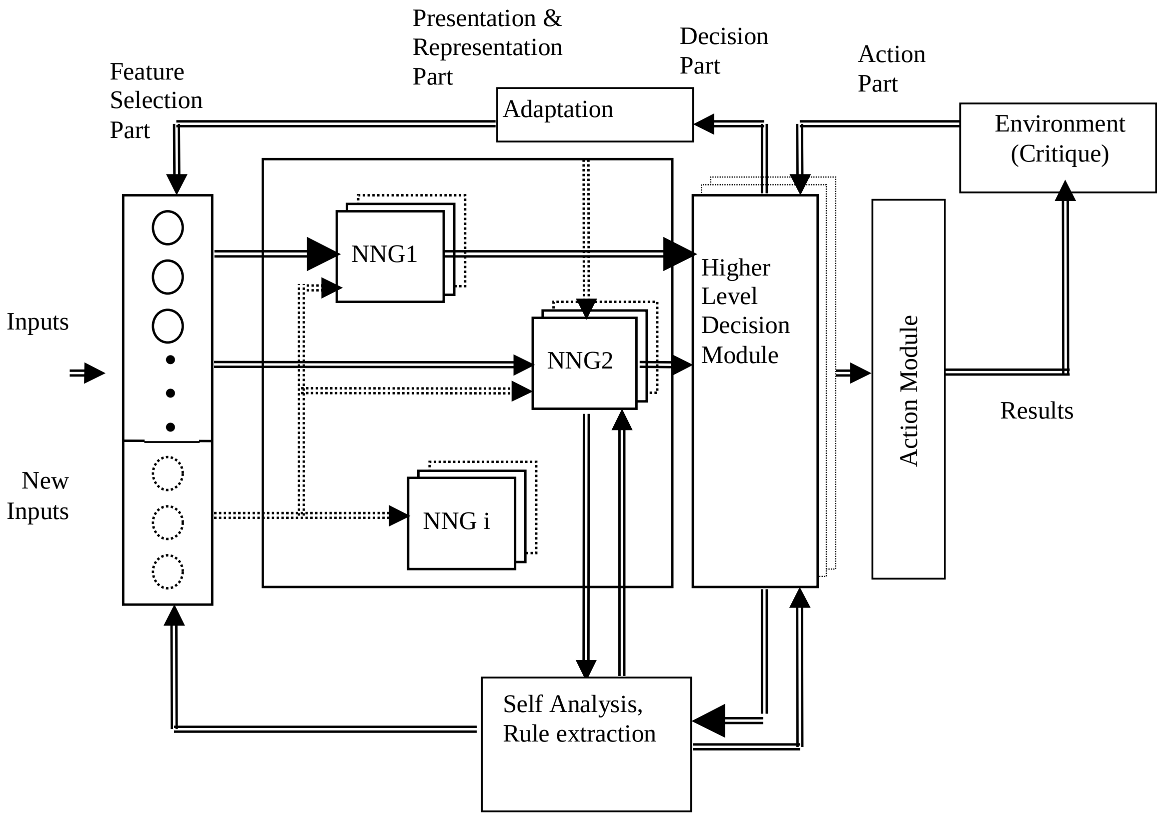
Experiment 1. **Incremental on-line learning.** Four EFuNNs are evolved with both positive and negative data. Sthr=0.9; Errthr=0.2; Person 1 EFuNN: rn=31 (8 positive); Person 2 EFuNN: rn= 35 (16 positive); Person 3 Efu NN: rn=35 (14 positive); Person 4 EFuNN: rn=29 (15 positive). Overall recognition rate: on training data - 11,16,17 and 20 (80% recognition rate); on test data: 2,2,6 and 7 (43%).

Experiment2. **Changing the number of the input variables**. Two time lags of 26-element MS vectors are added to the inputs and the FuNNs from experiment 1 are further trained with the new 78 element input vectors. Person 1 EFuNN: rn=56; Person 2 EFuNN: rn= 60; Person 3 EFuNN: rn=56; Person 4 EFuNN: rn=59; Overall recognition: on training data - 17, 20,20 and 20 (96.25% recognition rate); on test data: 7,2,2 and 8 (48%).

Experiment 3.**Sleep eco training.** First four EFuNNs are evolved with positive data only. Sthr=0.9; Errthr=0.2; Person 1 EFuNN: rn=15; Person 2 EFuNN: rn= 20; Person 3 EFuNN: rn=15; Person 4 EFuNN: rn=10. Overall recognition: on training data - 17,20,18,16 (89%); on test data: 7,2,2 and 6 (43%). After that eco training is applied. The recognition rate has improved to 96% on the training data and 53% on the test data.

4. CONCLUSION

ECOS have features that address the seven major issues from section one. The framework is currently applied to adaptive speech recognition, adaptive time series prediction, and integrated audio and video information processing.



References

[1] Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., Rosen, D.B., FuzzyARTMAP: A neural network architecture for incremental supervised learning of analog multi-dimensional maps, IEEE Trans. on Neural Networks,vol.3, No.5 (1991), 698-713

[2] Fritzke, B., Growing cell structures - a self- organising network for unsupervised and supervised learning, Neural Networks, vol.7, No.9 (1994) 1441-1460.

[3] Heskes, T.M., Kappen, B. On-line learning processes in artificial neural networks, in: Math. foundations of neural networks, Elsevier, Amsterdam, (1993)199-233

[4] Ishikawa, M., "Structural Learning with Forgetting", Neural Networks 9, 501-521 (1996).

[5] Kasabov, N. Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering, The MIT Press, CA, MA (1996).

[6] Kasabov, N., Postma, E., and Van den Herik, J AVIS: A Connectionist-based Framework for Integrated Audio and Visual Information Processing, in Proc. of Iizuka'98, Iizuka, Japan, Oct.1998

[7] Kasabov, N. Evolving Fuzzy Neural Networks - Algorithms, Applications and Biological Motivation, in Proc. of Iizuka'98, Iizuka, Japan, Oct.1998

[8] Le Cun, Y., J.S. Denker and S.A. Solla, "Optimal Brain Damage", in: Touretzky,ed., *Advances in Neural Inform.Proc.Systems*,MorganKaufmann,2,598-605(90).

[9] Quartz, S.R., and Sejnowski, T.J. The neural basis of cognitive development: a constructivist manifesto, Behavioral and Brain Science, to appear

[10] Reed, R., "Pruning algorithms - a survey", IEEE Trans. Neural Networks, 4 (5) 740-747 (1993).

[11] Watts, M., and Kasabov, N. Genetic algorithms for the design of fuzzy neural networks, in the same volume