A Predication Survival Model for Colorectal Cancer

PROF. DR. SHERIF KASSEM FATHY
Information System Department, College of Computer and Information Technology
King Faisal University
SAUDI ARABIA
Sherif_kassem@kfu.edu.com; Sherif_kassem@yahoo.com

Abstract: - The paper demonstrates an artificial neural networks (ANN) model for prediction survival of colorectal cancer. Data model is collected from SEER which is one of the largest and most comprehensive sources of information on cancer incidence and survival in the USA. Data set consists of over 100000 of colorectal cancer patients. Experimental results are carried out to get the minimum number of extracted features with an optimum ANN architecture without decreasing the prediction accuracy rate. Two models of prediction survival are described. In the first model, the experimental results show that the maximum prediction rate is 84.73%. In the second model, the main objective is to discover the minimum subset of input features that yields the highest accuracy. The experiment results reveal that 73.68% of selected features are sufficient for discrimination and the maximum prediction rate achieves 86.51%. Moreover, 72.72% of hidden neurons are sufficient to get optimum ANN architecture.

Key-words: Colorectal Cancer, Bio-computing, SEER, Artificial Neural Network (ANN), Feature extraction, Prediction rate

1 Introduction

Worldwide cancer is the second most common cause of death, exceeded only by heart disease. In the US alone, cancer accounts for nearly 1 of every 4 deaths. In 2009, the American Cancer Society estimated that about 562340 Americans were died of cancer [1]. That is, more than 1500 people would be died per day. Lung and Prostate cancer are the first and the second leading cause of cancer death in the USA. Colorectal cancer is the third leading cause of cancer death. Colon has four sections: ascending colon, transverse colon, descending colon, and sigmoid colon. Colorectal cancer is developing in the colon or the rectum. Colorectal cancer usually develops slowly over a period of many years. Before a true cancer develops, it usually begins as a noncancerous polyp which may eventually change into cancer. A polyp is a growth of tissue that develops on the lining of the colon or rectum.

On other hand, artificial neural networks, (ANNs) are regression devices containing layers of computing nodes (crudely analogous to the mammalian biological neurons) with remarkable information processing characteristics. They possess high parallelism, robustness, generalization and noise tolerance that make them capable of clustering, function approximation, forecasting, classification, prediction, pattern recognition and performing massively parallel multi-factorial analyses for modelling complex patterns where there is little a priori knowledge [2]. ANN models possessing such characteristics are desirable because: (a) nonlinearity allows better fit to the data, (b) noise-insensitivity leads to accurate prediction in the presence of uncertain data and measurement errors, (c) high parallelism implies fast processing and hardware failure-tolerance, (d) learning and adaptability permits the system to update and/or modify its internal structure in response to changing environment, and (e) generalization enables application of the model to unlearned data [3].

The main objective of this paper is to design and implement an ANN model that is capable for prediction survival of colorectal cancer. There are various literatures on cancer recognition and detection with a variety of methodologies that present results with various numbers of classes. We briefly describe some of these approaches. In [4], the paper described building an ANN model and regression tree (CART) for colorectal cancer patients. The paper showed that linear correlation coefficients

ISBN: 978-960-474-270-7

were high in both models and the mean absolute errors were similar. But, ANN models demonstrated a better linear correlation than CART model. In [5], data set of the American College of Surgeons' Patient Care Evaluation (PCE), Department of Medicine, New York Medical College, New York, contained only the TNM variables (tumour size, number of positive regional lymph nodes, and distant metastasis). The experimental results showed that the artificial neural network's predictions of the 5-year survival of patients with breast carcinoma were significantly more accurate than those of the TNM system.

Nowadays, gene expression data is being increasingly utilized for cancer detection. In [6], the gene was inputted to ANN for prediction the survival rate. In [7], the paper proposed an evolutionary neural network that might be able to classify the gene expression profiles into normal or colon cancer cell. Experimental results on colon microarray data demonstrated that the proposed method was better than other classifiers.

2 Data Analysis

2.1 Data Source and Size

In this paper, data set is obtained from SEER [1] Program of National Cancer Institute. SEER program currently collects and publishes cancer incident and survival data from 14 population-based cancer registries and three supplemental registries covering approximately 26% of the USA population. Quality control has been an integral part of SEER since 1973. Every year, studies are conducted in SEER areas to evaluate the quality and completeness of the data being reported. Information on more than three million in situ and invasive cancer cases is included in the SEER database.

2.2 Data Filtering and Validation

Every year SEER provides the end user with various ASCII files. Each file is dedicated for specific cancer type. A patient profile is presented by a single record with more than 70 variables. These variables may be utilized as parameters or features. We only concentrate on the colon cancer files from 2004 to 2007 for the following reasons: older files had different record format structure and contained missing data, or unknown values. In this research, data filtering and validation are performed. Data filtering and validation process indicate that more than 37% records are inadequate. Inadequate records are removed from the real data sample. This results in reduction of the data set to over 100000 records.

2.3 Feature Selection

There are more than 70 features in SEER documents [8]. It is out of scope of this research to describe the physical meaning of each feature. For more details and interpretations of each feature, refer to SEER documents. We only select 20 features, since the other variables are not related to our work.

Table 1. The Selected Feature of Colorectal Cancer

Feature	Feature Interpretation
F01	Number of counted Tumours
F02	Laterality (Tumours Position)
F03	Grade
F04	Diagnostic Confirmation
F05	Primary Tumour Size
F06	Largest Tumour Size
F07	Lymph Node Chain
F08	Regional Nodes Positive
F09	Regional Nodes Examined
F10	Tumour Extension
F11	Historic Stage A
F12	Radiation
F13	Radiation Sequence with Surgery
F14	Histological Type
F15	Marital Status
F16	Race
F17	Sex
F18	Age
F19	Birth Place
F20	Survive

Table-1 shows the selected features. SEER defines a variable data field called "Survival Time Recode" or "Survivability".

ISBN: 978-960-474-270-7

"Survivability" indicates that an individual has been diagnosed with cancer and living for 60 months or more after. So, a new external variable is defined (called survive) and is set to one (survive = 1), if a patient is surviving more than 60 months, otherwise (survive = 0).

2.4 K Fold Cross Validation Model

In the machine learning environment, the performance of a classifier is usually measured in terms of prediction error. In most real-world problems like ANN, the error cannot be exactly calculated and it must be estimated. Therefore, it is important to choose an appropriate estimator of the error. Although, K-fold cross validation is an old model, it is still used nowadays [9,10] to improve the selection of test data set. The data set is divided into k subsets and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then, the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set k-1 times. The variance of the resulting estimate is reduced as k is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch k times which mean it takes k times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set k different times. The advantage of doing this is that we can independently choose how large each test set is and how many trials we average over.

3 ANN Architecture

In this work, the back propagation ANN architecture is utilized for prediction of survival of colorectal cancer. The basic structure of the neural network is a standard three-layer feed forward network which consists of an input layer "I", a hidden layer "H", and an output layer "0". Simply, the ANN can be described as [I,H,O] structure. The ANN back propagation architecture depends upon selection of both appropriate numbers of features and architecture. The number of inputs "I" corresponds to the dimensionality of the input features (19 features). There is only one output (F20) that indicates either survival or not.

However, there are various problems that have to be solved. Some of these problems are: which training algorithms are suitable for it and how to connect the new added components in neural networks. [11,12] suggested that the use of single hidden layer was fully justifiable in view of network ability to accurately approximate arbitrary functions, provided a sufficient number of hidden nodes even though there was a potential to use flexible neural network architecture as many layers and as many hidden units as were required.

There are two approaches to get the optimum number of nodes in the hidden layer(s) of the neural network. The first approach is to start with a small neural network [13] and iteratively increase the number of nodes in the hidden layer(s) until satisfied learning occurs. The second approach is to begin with a large network and make it smaller by iteratively eliminating nodes in the hidden layer(s). These types of algorithms are called pruning algorithms [14-18].

3.1 ANN Implementation, Training and Testing

In this research, ANN is fully implemented using C++ language. No especial package is used. For each experiment, each network is trained using the back-propagation algorithm with a momentum term. A sigmoid function is utilized as activation function for all hidden nodes of the generated networks. All architectures are fully connected with one hidden layer. All connection weights and nodes threshold values are randomly initialized. These values are uniformly distributed between (0,1). To guarantee equal effect of input features to the neural network, the following general normalization equation is used to normalize inputs to be between (0, 1).

$$z_i = \frac{1.0}{1.0 + e^{-\left(\frac{x_i - \mu_i}{2.0\,\sigma_i}\right)}}$$

where $Z_i \in (0,1)$ is the corresponding scaled input value of the original input value X_i, σ_i is

the standard deviation and μ_i is the mean value of input feature X_i.

The experimental data is divided into 10 sets. All experiments are performed using the K Fold cross validation model. So, there is no overlap between the training and testing sets.

The ANN is trained for 10 times for given hidden units in order to avoid the randomization effect of the weights. For analysis purpose, the accuracy of ANN is taken as the mean values of all the 10 training. The training continues until the Root Mean Square Error (RMSE) or validation error fails to decrease by a certain amount of error over a given period. Each network is monitored for the improvement of every 500 epochs to justify the continuation of the training.

4 Experimental Results

In ANN, as stated before, both feature extraction process and architecture design process are very critical. Extensive independent investigations of either of the process alone may not give an optimum result [19,20]. So, the paper describes two models to get the suitable number of extracted features with an optimum ANN architecture without decreasing the prediction accuracy rate. The ANN architecture may be considered optimum, when it contains only the minimum number of hidden neurons and minimum number of inputs (features).

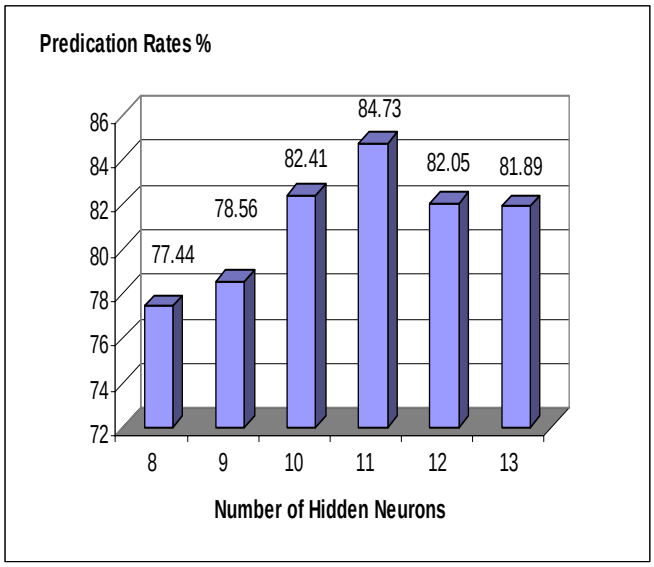
4.1 First Model

ANN begins with [19-h-1] structure, where h = 13, 12, ..., 8. That is, the experiment starts with suitable number of hidden nodes and iteratively eliminating nodes in the hidden layer using pruning algorithm [14-18]. In each case, the experiments are carried out using K fold model and the average is taken.

Fig-1 illustrates number of hidden neurons versus prediction rates using the K fold model. The accuracy of the ANN is measured using prediction error based on Root Mean Square Error (RMSE).

Fig-1 shows that predication rate ranges between 77.44% and 84.73%. Fig-1 illustrates that although when the number of hidden neurons is reduced from 13 to 11, the prediction rate is increased from 81.89% to 84.73%. So, the highest prediction rate can be achieved with 11 hidden neurons. Many investigators, in the artificial intelligence field, note that most of the times ANNs may offer better predictive ability but not much explanatory value. This criticism is generally true. Recently, it has become a commonly used method in ANN studies for identifying the degree to which each input channel (independent variables or decision variables) contributes to the identification of each output channel (dependent variables).

Fig 1 Number of Hidden Neurons versus Predication Rates



4.2 Second Model

The goal of the second model is to find the minimum subset of input features that yields the highest accuracy. Input features may be classified into two categories: effective features and non-effective features. Identifying the effective features is not an easy task. This problem is especially severe when real-world applications are attempted. There are two models of features subset selection [21]: filter model and wrapper model. In filter model, the features are filtered independently of the induction algorithm. This filtering is done as a pre-processing step. In contrast, wrapper model wraps around the induction algorithm, searching the feature subset space guided by the performance of the induction algorithm.

ISBN: 978-960-474-270-7

Since the filtering model ignores the effect of the feature subset on the performance of the induction algorithm, many researchers have pointed out that it may not be as effective and general as the wrapper model [21-23]. In the wrapper model, a large number of training is required to search for the best performing features subset; it can be prohibitively expensive for neural net applications. Many research strategies were proposed to speed up the search. In [24], features subset search is accelerated by a heuristic Artificial Neural Net Input Gain Measurement Approximation (ANNIGMA). ANNIGMA ranks neural network net features by relevance. So, a huge improvement in speed may be achieved.

i ANNIGMA Algorithm

Detail describing of ANNIGMA is out of scope of this paper. The basic concepts with main equations will be described only. For more details, refer to the original paper [24].

ANNIGMA ranks neural network net features by relevance based on the weights associated with the features, since the weight in ANN can be viewed as representing the gain of the input signal to the output node. Input signals that are noisy or irrelevant to output will have a high error if they have high associated weights. Therefore, training algorithms must reduce their weights such that they do not contribute to the output. In similar manner, the weights of relevant and noise-free signals will be increased.

ANNIGMA introduces the total relative gain "G" of a particular input signals "i" to a particular output node "o" by:

$$G_{io} = \sum_h | W_{ih} * W_{ho} |$$

where i, h, o are the input, hidden, and output layer indices respectively. Then, the gain "G" of any particular input node "i" is normalized to a scale 100 called Normalized Gain (NG). NG can be defined as:

$$NG_{io} = (G_{io} / max_i(G_{io})) * 100$$

So, the basic idea is that the ANN weight of each input feature is monitored and the normalized gain is calculated. It is clear that the inputs with the largest normalized gain values, in ANN, highly affect the output values. In our case, as shown in the first model, the optimum ANN architecture is [19-11-1]. The normalized gains of all 19-input features are calculated using the K fold model and the averages are calculated. Fig-2 shows input features versus normalized gains.

Fig 2. Input Features versus Normalized Gains

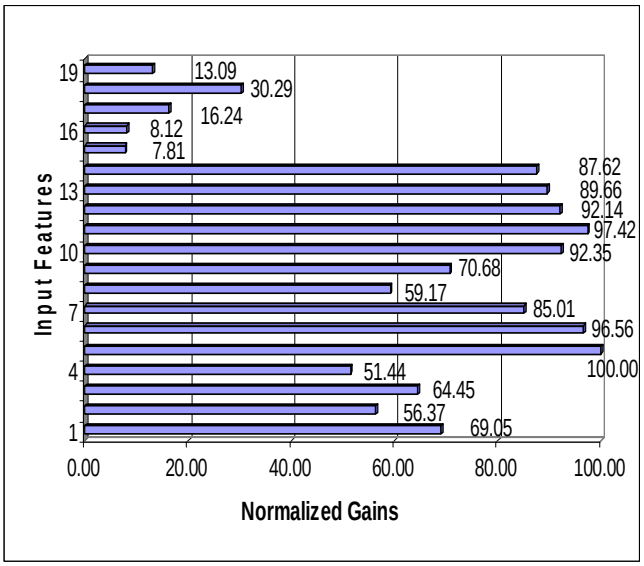
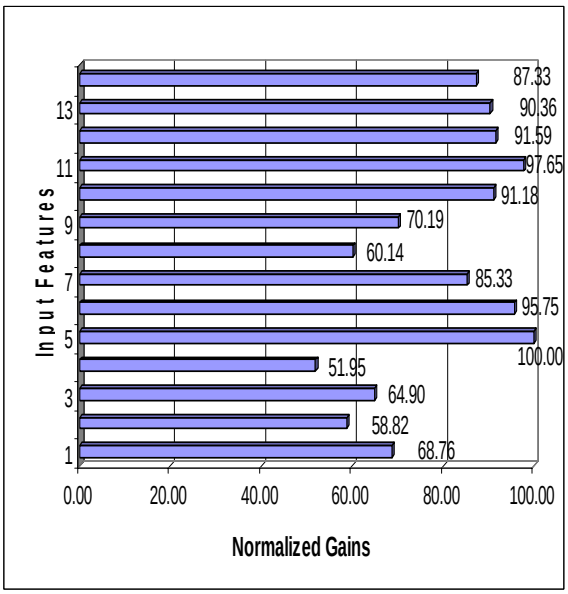


Fig-2 illustrates that F05 (Primary Tumour Size), F11 (Historic Stage A) and F06 (Largest Tumour Size) have the highest gain values. F15 (marital status), F16 (Race), F17 (sex), and F19 (Birth Place) have the lowest gain values. While F18 (Age) has intermediate gain values.

Fig 3. Input Features versus Normalized Gains after Elimination



ii Eliminating Features Model

In this work, eliminating features model is based on the backward strategies. F15 with minimum normalized gain is eliminated from the input features. Then, new normalized gains are calculated with the remaining 18 input features. This process is repeated until the Root Mean Square Error (RMSE) does not decrease to certain limit. Fig-3 shows input features versus normalized gains after four iterations. Only F1: F14 are utilized in second model and other features with minimum Normalized Gains are excluded.

iii Optimum ANN Architecture

The same algorithm in first model is repeated with a new ANN [14, h, 1] architecture where h= 10, 9, ... , 5. In each case, the experiments are carried out using K fold model.

Fig-4 shows number of hidden neurons versus new prediction rates using the K fold model. The accuracy of the ANN is measured using Root Mean Square Error (RMSE).

Fig 4. Number of Hidden Neurons versus New Predication Rates

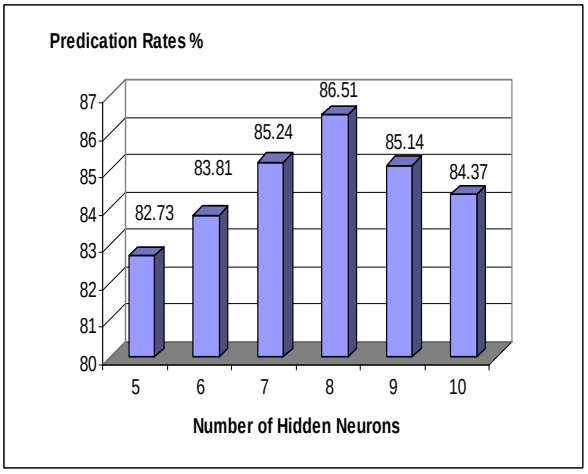


Fig-1 and Fig-4 show that the new predication is increased by 1.78%, although the numbers of features are reduced from 19 to 14 features and the numbers of hidden neurons are reduced from 11 to 8 neurons.

5 Conclusion

In ANN, both feature extraction process and architecture design process are very critical. Extensive independent investigations of either of the process alone may not give an optimum result. So, the paper describes methodology to get suitable number of extracted features with an optimum ANN architecture without decreasing the prediction accuracy rate. In the second model, the ANN architecture is optimum since it contains only the minimum number of inputs and hidden neurons. The proposed optimum ANN architecture is tested for predication survival of colorectal cancer problem. The ANN is trained iteratively until certain criteria are met. The hidden layer in ANN architecture is changed according to the training test data using Minimum Root Mean Square Error. ANN is trained using K fold cross validation model to avoid the randomization effect of the weights and minimize Root Mean Square Error. In the first model, the experimental results show that the maximum prediction rate is 84.73% with 19 features.

In the second model, the main effective features are discovered to eliminate the irrelevant and redundant features. Effective features can be reduced to 14 features only without decreasing the Minimum Root Mean Square Error. That is, only 73.68% of the features are quite enough for prediction. Moreover, 72.72% of hidden neurons are sufficient to get optimum ANN architecture.

It has to be mentioned that the second model has a higher prediction rate than the first model, although the number of features in the second model is less than the number of features in the first model. That is because; we concentrate on the effective features only using ANNIGMA algorithm. Non-effective features may be considered as a noise in ANN. The paper shows that both ANN and cancer data are overwhelming channels for further researches.

References

[1] SEER Publication, Cancer Facts, 2009, http://seer.cancer.gov/publications/.

[2] Simon Haykin, Neural Networks and Learning Machines, 3rd Edition, 2008.

[3] Bishop C.M., Neural Networks for pattern recognition, Oxford University Press, New York, 1997.

[4] Seung-Mi Lee, Jin-Oh Kang, Yong-Moo Suh, Comparison of Hospital Charge Prediction Models for Colorectal Cancer Patients: Neural Network vs. Decision Tree Models, The Korean Academy of Medical Sciences, ISSN 1011-8934, 2004.

[5] Harry B. Burke, Artificial Neural Networks Improve the Accuracy of Cancer Survival Prediction, the annual meeting of the American Joint Committee on Cancer, Scottsdale, Arizona, 1995.

[6] Farid E Ahmed, Artificial neural networks for diagnosis and survival prediction in colon cancer, 2005. http://creativecommons.org/licenses/by/2.0

[7] Kyung-Joong Kim, Sung-Bae Cho, Prediction of colon cancer using an evolutionary neural network, Neurocomputing 61, pp. 361-379, 2004.

[8] SEER Publication, Research Data Record Description Cases Diagnosed, 2009, http://seer.cancer.gov/publications/

[9] R. Kohavi, A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, Proc. Int'l Joint Conf. Artificial Intelligence, pp. 1137-1145, 1995.

[10] Juan Diego Rodríguez, Aritz Pérez, and Jose Antonio Lozano, Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32 no. 3, pp. 569-575, 2010.

[11] K. Homik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, Dep. Of Economics, University of California, San Diego, manuscript, 1988.

[12] R. Hecht-Nielson, Theory of back propagation neural network, Proc. Of Int. Joint Conf. on Neural Networks, New York; Wiley, 1989.

[13] Hirose, Y., Yamashita, K., and Hijiya, S., Back-Propagation Algorithm Which Varies the Number of Hidden Units, Neural Networks, Vol.4, pp. 61- 66, 1991.

[14] Sietsma, J., And Dow, R.J.F., Neural Net Pruning: Why and How, In Proceedings of IEEE Int. Conf. Neural Networks, San Diego, Vol.1, pp. 325-333, 1988.

[15] Reed, R., Pruning Algorithms - A Survey, IEEE Transactions on Neural Networks, Vol.4, No.5, pp. 740-747, 1993.

[16] Karnin, E.D., A Simple Procedure for Pruning Back-Propagation Trained Neural Networks, IEEE Transactions on Neural Networks, Vol.1, No.2, pp. 239-242, 1990.

[17] Dreyer. P., Classification of Land Cover Using Optimized Neural Nets, Photogrammetric Engineering & Remote Sensing, Vol.59, No.5, pp. 617- 621, 1993.

[18] K. Mohraz and P. Protzel, A Flexible Neural Network Construction Algorithm, Proc. of the 4th European symposium on Artificial Neural Networks (ESANN'96), Brussels, 111-116, 1996.

[19] J. M. Steppe, K. W. Bauer and S. K. Roger, Integrated feature and architecture selection, IEEE Trans. Neural Networks, 7(4), 1007-1014, 1996.

[20] H. I. Avi-Irzhak, T. A. Diep, and H. Garland, High accuracy optical character recognition using neural networks and centroid dithering, IEEE Trans. Pattern Anal. Machine Intel, 17, 218-224, 1994.

[21] G.H. John, R. Kohavi, and K. Pfleger, Irrelevant features and the subset selection problem, in Machine Learning: Proceedings of the Eleventh International Conference (ICML '94), San Francisco, CA, 1994.

[22] R. Kohavi and D. Sommerfield, Feature subset selection using the wrapper methods: Overfitting and dynamic search space topology, in Proceedings of the first international conference on Knowledge Discovery and Data Mining, KDD '95, Menlo Park, CA, 1995.

[23] R. Carunan and D. Freitag, Greedy attribute selection, in Machine Learning: Proceedings of the 9th International Conference on Industrial and Engineering Applications of AI and ES, 1996.

[24] Chun-Nan Hsu, Hung-Ju Huang, and Dietrich Schuschel, The ANNIGMA-Wrapper approach to fast feature selection for neural nets, IEEE transaction on system, man and cybernetics-part b: cybernetics, 1999.