Radial Basis Functions Networks to hybrid neuro-genetic RBFNs in Financial Evaluation of Corporations

Loukeris Nikolaos

Abstract-Financial management maximise investors' return, seeking for stocks with increasing expected corporate value. Hidden information is included in vast accounting data and financial indices that are available in international financial markets. Methods of Econometrics and Artificial Intelligence- mainly in the field of Neural Networks- provide classifications of companies regarding their economic health. Radial Basis Function networks are examined in a hybrid form of Neural Network optimised with Genetic Algorithms and in a regular Neural Net form, to determine efficient methods of Financial Analysis. The regular Radial Basis Function network with 3 layers, Genetic Algorithms in all the layers and Cross Validation is superior to all the neuro-genetic forms of RBF in Financial Analysis.

Keywords- Accounting, Corporate Finance, Genetic Algorithms, Radial Basis Function Networks

I. Introduction

FINANCIAL analysis requires precise evaluations of corporations to support decision making in investment portfolios of mutual funds or stocks. Valuable information of critical importance is included in accounting statements of corporations. The complexity of accounting data, and their extended form that is usually met in time series form, requires effective methods to extract hidden knowledge from vast data acquiring significant information that may return significant profits to investors, forming lucrative portfolios. Linear models have been used in econometrics quite long time, whilst neural networks offer a new aspect in the same methodology, Orr (1996).

Radial Basis Function-RBF networks of supervised learning are a specific type of linear model, capable to perform regressions, classifications and time series predictions. Estimation of unknown functions in statistics, from a set of input and output data with no previous knowledge of the function, used techniques of: nonparametric regression, function approximation, inductive learning, being identical to the supervised learning of neural networks. The training set, equivallent to the set of examples, contains values of the independent (input) variable, sensitive to noise, and the dependent (output) variable. Regressions are either parametric, where the relation between the dependent and independent variables is described by a known function, including unknown parameters, or nonparametric with no previous knowledge of the function under estimation implementing an equation with free parameters and permitting a vast variety of functions to be approached.

Neural networks are nonparametric models and many of their elements, such as the weights have no physical meaning. The basic target of neural networks is the estimation of the requested function. Classification aims to categorize hidden patterns to their respective classes based on previous examples from each class, producing a discrete output, in contrast to the non parametric regression that produces continuous values. Linear models have the form: $f(x) = \Sigma^m w_j h_j(x)$, where $f(x)$ a linear combination of a set of m fixed functions-the basis functions, w_j the coefficients of linear combinations and h_j the equivellance of weights and hidden layers to basis functions.

The function *f*, is flexibile, fitting many different functions, and its weights can choose values freely. The basis functions and any parameters which they might contain are fixed, but when the basis functions can change during the learning process, the model is nonlinear. Any function type can be used as the basis set, whilst differentiable functions are prefered. Basis functions are popular in polynominal form such as $h_i(x)=x^i$, whilst logistic functions as: $h(x) = 1/[1+exp(b^Tx-b_0)]$ are prefered in Multi Layer Perceptrons neural networks. The least square error can solve supervised learning problems providing the optimal weight values implied by the training set. Nonlinear models, such as MLPs, require iterative numerical procedures for their optimisation.

Radial functions are functions whose the response change monotonically (decrease or

Manuscript Received April 10 2008; Revised received July 17, 2008
Loukeris Nikolaos is with the *University of Essex, Email: nikosloukeris@gmail.com, nlouke@essex.ac.uk*

increase) with distance from a central point, such as the Gaussian function. The parameters of the model are the centre, the distance scale, and the shape of the radial function. The Gaussian RBF is: $h(x) = exp [-(x-c)^2/r^2]$, and decreases monotolically from the centre, whilst the multiquadric RBF: $h(x) = \sqrt{[r2+(x-c)^2]}/r$ increases monotonically with distance from the centre. Radial functions are used in linear or nonlinear models and in networks with a single or multiple layers, whilst an RBF network is nonlinear if the basis functions can move or change size or if there is more than one hidden layer. Radial Basis Function networks –RBF have been associated with radial functions in a single-layer network, Broomhead and Lowe (1988). A weight penalty term added to the sum-squared-error, as in ridge regressions, the cost function: $C = \Sigma_{i=1}(\hat{y}_i - f(x_i))^2 + \Sigma_{j=1} \lambda_j w_j^2$ is minimised, where $\{\lambda_j\}^m{}_j = 1$ regularisation parameters. Minimisation of the cost function leads to a set of *m* simultaneous linear equations in the *m* unknown weights. The model selection criteria are estimates of prediction error, indicating the efficiency in which the trained model will perform on unknown future inputs. The best model is the one whose estimated prediction error is least.

Cross-validation is the standard tool for measuring prediction, whilst an alternative form is given by generalised cross-validation, creating many adjustments to the sum-squared-error over the training set. Available input and output measurements are divided into two parts - one for training and another part for testing given that data is not scarce. Thus different models trained on the training set, can be compared on the test set, consisting the basic form of cross-validation. Weight decay pruned unimportant network connections, making necessary to add the same penalty term to the sum-squared-error as in ridge regression, Orr (1996). Mean-squared-error between the input x, the true output y(x), and the output prediction of the trained model is: $MSE = \langle(y(x) - f(x))^2\rangle = (y(x) - \langle f(x)\rangle)^2\rangle + \langle(f(x)) - \langle f(x)\rangle)^2\rangle$ where the expectation $\langle(y(x) - f(x))^2\rangle$ is taken over the training sets, $(y(x) - \langle f(x)\rangle)^2\rangle$is the bias and $\langle(f(x)) - \langle f(x)\rangle)^2\rangle$the variance. For $\langle(f(x)\rangle = y(x)$ to all x the model is unbiased, possibly with large mean-squared-error if it has a large variance, indicating that f(x) is highly sensitive to the peculiarities of each particular training set causing, regression problems to be ill-posed, Tikhonov (1977). The variance may be significantly reduced, introducing a small amount of bias causing a reduction in mean-squared-error, which restricts the range of functions to the model by removing degrees of freedom.

II. Radial basis functions

The most general form for RBF is: $h(x) = \varphi[(x-c)^T R^{-1}(x-c)]$ where φ is the function used (Gaussian, multiquadric), c the centre and R the metric. The $(x-c)^T R^{-1}(x-c)$ is the distance between the input x and the centre c in the metric defined by R. Many types of functions are used: the Cauchy $\varphi(z) = (1+z)^{-1}$, the Gaussian $\varphi(z) = e^{-z}$, the multiquadric $\varphi(z) = (1+z)^{1/2}$, and the inverse multiquadric $\varphi(z) = (1+z)^{-1/2}$. Least squares applied to supervised learning with a linear model, minimize the function of sum-squared-error : $S = \Sigma^{\mu} [\hat{y}_i - f(x_i)]^2$ where $f(x_i) = \Sigma_{j=1}{}^m w_j h_j(x)$, the free variables are the weights $\{w_j\}^m = 1$. Radial Basis Functions networks implement regularization theory to ill-conditioned problems. In problems of revealing the map that transforms input samples into a desired classification, having only a few samples, quite noisy, the problem is ill-posed. The decrease of error between the network output and the desired response can provide solution, including an added constraint of smoothness. A Radial Basis Function-RBF network is in figure 4: where every input component (p) is to a layer of hidden nodes, each node in the hidden layer is a p multivariate Gaussian function (radial basis function): $G(x, x_i) = \exp\{[-1/2\sigma_i^2] \Sigma(x_k - x_{ik})^2\}$ of mean x_i and variance, the effect of linear weight to the hidden nodes, produces the output that may lead to a very large hidden layer: $F(x) = \Sigma^N w_i [G(x, x_i)]$.

This solution is approximated by reducing the number of Processing Elements in the hidden layer, and placing them over the input space regions. The positions of each radial basis function and its variance are estimated, with the linear weights. The unsupervised learning rule of k-nearest neighbour is vastly used to estimate the centres and widths. The input space is discretized into k clusters, (its size is obtained from input data). Clusters centres give the centres of the RBF network, and the distance between clusters is the width of the Gaussians. Through competitive learning the centres and widths are found, each width is set proportional to the distance between the centre and its nearest neighbour. The output weights are obtained through supervised learning, the output unit is normally linear and convergence is faster. Usually, a Multi Layer Perceptron-MLP takes advantage of nonlinearly separable data clusters produced by very few RBF, and consequently MLP is more efficient than the linear network.

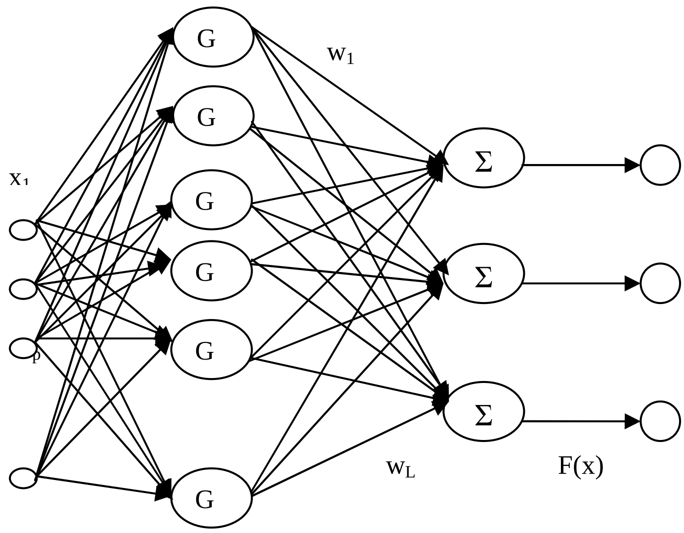


Figure 1. A Radial Basis Function network

II.I RBFNs with Genetic Algorithms optimisation

Radial Basis Function (RBF) networks-figure 4, 5- are nonlinear hybrid networks, under certain circumstances, typically containing a single hidden layer, or more, of processing elements. This layer uses Gaussian transfer functions, rather than the standard sigmoidal functions employed by MLP. The centres and widths of the Gaussians are set by unsupervised learning rules, whilst supervised learning is applied to the output layer. These networks tend to learn much faster than MLPs.

Some other types of RBF networks are the Generalized Regression Neural Network-GRNN, the Probabilistic Neural Network - PNN, where all the weights can be calculated analytically when they are applied, figure 6, and the number of cluster centres is by definition equal to the number of exemplars, being all set to the same variance. These special types of RBF are used only when the number of exemplars is of a low quantity, less than 100, or so dispersed that clustering is ill-defined.

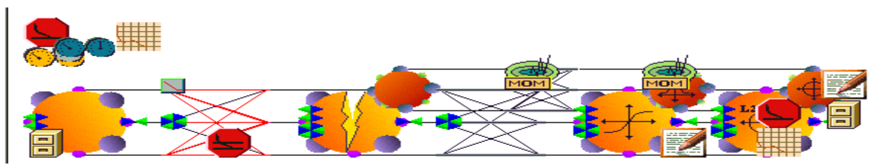


Figure 2. The RBFN in laboratory, Source: NeuroDimensions Inc.

In regular RBF networks, the supervised segment produces a linear combination of the output at the unsupervised layer, and by default there are 0 hidden layers. Hidden Layers can be added to make the supervised segment a MLP instead of a simple linear Perceptron. The amount of Gaussians, which cannot be predefined and are problem dependent, is entered using the Cluster Centres field. The number of patterns in the training set affects the number of centres, depending on the dispersion of clusters, where in case data are well clustered, few Gaussians are needed. In case data are there should be more Gaussians to acquire a fine performance. Competitive learning is used offering the choice of the dot product metric, which measures the angle between the present input and the weight vector, or the Euclidean metric, that measures the difference between the two vectors maintaining distances in the input space. When inputs and weight vectors are normalized, the two metrics are equivalent. Competitive learning has an intrinsic probability distribution of the input data, where a number of PE is possible to be idle, while others may always win the competition. A conscience mechanism that counts the frequency of each PE that wins the competition, and enforcing a constant wining rate across the PEs. The unsupervised segment of neural hybrid is trained independently from the supervised one. Termination of training procedure in the unsupervised segment of the network is based on the maximum number of epochs that is selected, where a high value ensures a matured processing time, and the unsupervised training is ended according on the change of weights. Thus training will end when all the weights change by less than the specified value from one epoch to the next. In laboratory unsupervised learning usually performed best when the learning rate started out high having a gradual decay during training, permitting the network to find an approximate solution quickly, and to focus on details of the problem afterwards, Principe, Euliano, and Lefebvre (2000).

The importance of each one of the 16 financial inputs in hybrid RBF network is not predefined thus Genetic Algorithms select the significant inputs. The network must be trained multiple times to find the inputs combination with the lowest error. Genetic Algorithms were used on each layer in RBF with different topologies. On-Line learning were chosen to update the weights of hybrid neuro-genetic RBF, after the presentation of each exemplar. Genetic Algorithms were used providing solution to the problem of optimal values in a) Processing Elements, b) Step Size and c) Momentum Rate. RBF network must be trained multiple times to achieve the settings that result the lowest error. Output layer was chosen to implement Genetic Algorithms optimizing the value of the Step size and the Momentum. Data came by 1411 companies from the loan department of a Greek commercial bank, with the following 16 financial indices:

1) EBIT/Total Assets**,**
2) Net Income/Net Worth,
3) Sales/Total Assets,
4) Gross Profit/Total Assets,
5) Net Income/Working Capital,
6) Net Worth/Total Liabilities,
7) Total Liabilities/Total assets,
8) Long Term Liabilities/(Long Term Liabilities + Net Worth),
9) Quick Assets/Current Liabilities,
10) (Quick Assets-Inventories)/Current Liabilities,
11) Floating Assets/Current Liabilities,
12) Current Liabilities/Net Worth,
13) Cash Flow/Total Assets,
14) Total Liabilities/Working Capital,
15) Working Capital/Total Assets,
16) Inventories/Quick Assets,

and a 17th index with initial classification, done by bank executives. Test set was 50% of overall data, and training set 50%.

A variety of hybrid combinations was chosen to detect the performance of RBF Networks: i) RBF Nets with no GA, ii) RBF Nets with GA in input layer only,
iii) RBF Nets with GA in all layers, vi) RBF Nets with GA in all layers and Cross Validation

III. Results

In NeuroSolutions the RBF Neural Net, had fast convergence times near half a minute, MSE of a quite medium level, and a low positive correlation coefficient r, with equivalent variation between the inputs and an adequate fitness of the model to data. The optimal net was the RBF with 1 hidden layer where, healthy companies were identified as healthy at 98.65%, and the distressed as distressed at 31.19%, the cost (MSE) was low at 0.300 and r at 0.538 was adequate signifying low partial correlation between the variables, with an inadequate fitness of the model to the data, whilst it converged fast at 31 seconds.

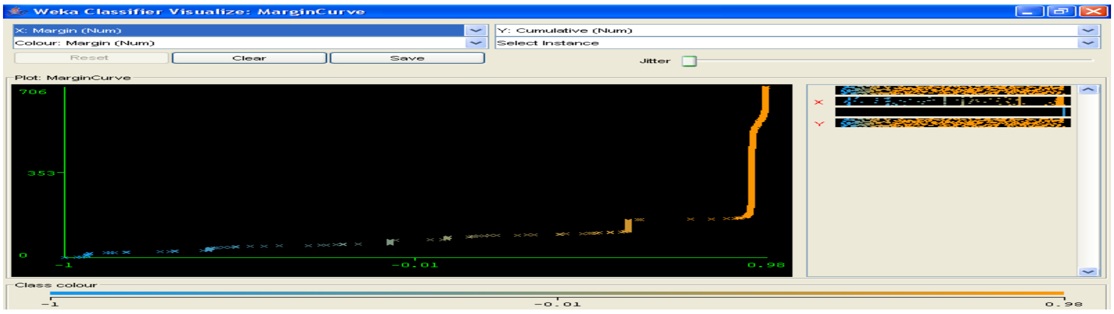
Regarding the Hybrid RBF models, the RBF with GA in the input layer only, wad an optimal result in a net of 2 hidden layers where 97.15% of healthy companies were classified as healthy, and 77.98% of the distressed as distressed, the cost described by Mean Square Error indicates an acceptable fitness of the network output to the desired output, was quite low at 0.202, with a strong correlation between inputs at 0.748, at a significant computing time of 6 hours 29 min. 43 sec.

Hybrids with GA in all the hidden layers had the optimal net at 2 hidden layers where 96.48% of the healthy companies were classified as healthy, and 76.14% of the distressed as distressed, the MSE cost was very low at 0.161, and was r appropriately high at 0.787, but the processing time was extended at 9 hours 34 min. 26 sec. A faster but inferior solution was given by the hybrid net of no hidden layers at 5 hours 20 min. 30 sec. with slightly higher results for the classified healthy companies at 97.31% and lower at 68.8% for the distressed.

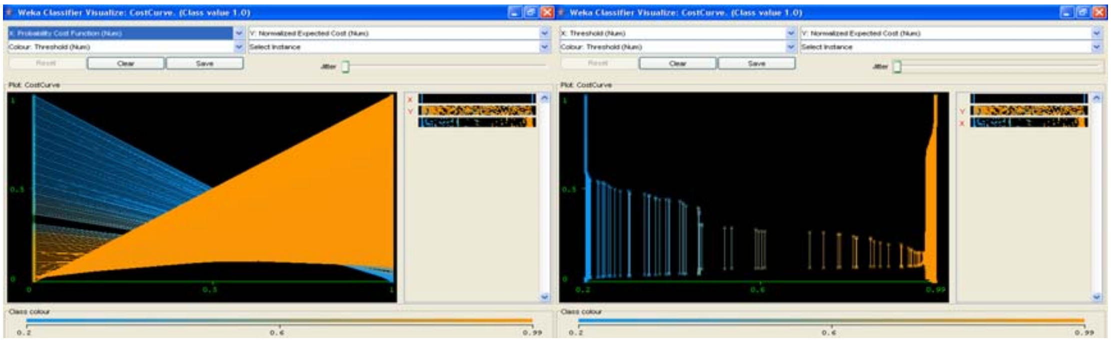
Hybrid RBFs with GA in all layers and Cross Validation had the optimal result with 3 hidden layers where healthy were classified successfully at 98.15%, and 73.39% of the distressed were identified as distressed firms, the error cost (MSE) was very low at 0.144 with a high correlation quality of the data in the same direction at 0.812, whilst the computation al time was extended at 9 hours 39 min. 44 seconds, as the Cross Validation results were similar. A quite close result was given from the hybrid net of 1 layers but the cross validation results were quite lower, and the convergence time was 2 times higher.

The overall evaluation of the RBF NN and the hybrid RBF nets revealed that the RBF with 3 layers GA in all and CV was the optimal net with an extended processing time of 9 hours 39 min. and 44 sec., whilst a faster and quite good solution can be given from an RBF with GA on the input layer only, having a slightly higher MSE cost at 0.202 in 6 hours 29 min. and 43 sec. A third solution can be the hybrid RBF with GA in all layers which offers results similar to the RBF with GA and CV but with slightly higher MSE and lower correlation coefficient, in the same processing time. The RBF NN was the fastest network but with the highest MSE cost, and the lowest r.

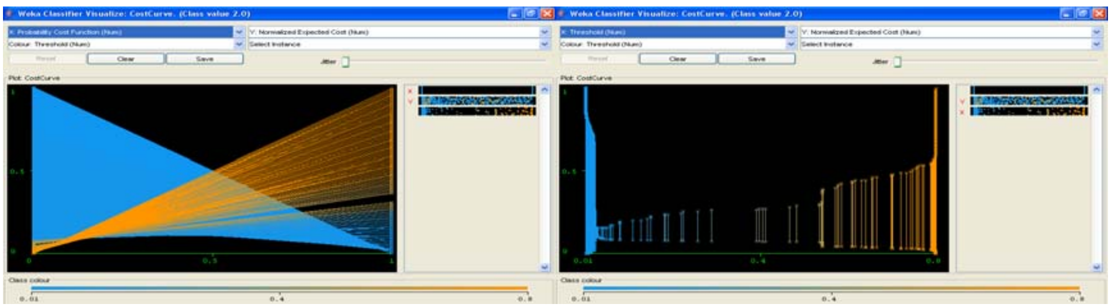
The Radial Basis Function neural network performed worst in the WEKA data mining platform as a neural network, since it converged very fast in 0.69 seconds. It uses the k-means clustering algorithm to provide the basis functions and learns either a logistic regression (discrete class problems) or linear regression (numeric class problems) on top of that. Symmetric multivariate Gaussians are fit to the data from each cluster. If the class is nominal it uses the given number of clusters per class. It standardizes all numeric attributes to zero mean and unit variance. The RBF network classified 77.9% of the initially characterized companies as healthy by loan experts in the category of the healthy companies, 2.97% of the initial healthy companies were classified as in distress, 7.93 of companies in distress were classified as healthy, whilst 11.1% of companies categorized as in distress by bank executives, were put in the class of in distress companies. The cost expressed as the Mean Square Error was very low 0.0748 indicating a very satisfactory fitness of the network output to the desired output.



Graph 1. The Margin Curve for a regular RBF network



Graph 2. The cost curve, in Probability Cost Function (left), and Threshold (right) for class 1



Graph 3. The cost curve, in Probability Cost Function (left), and Threshold (right) for class 2

Table 1. Results of hybrid RBFN's with Genetic Algorithms, in NS software

Neural Network		Active Confusion Matrix				Performance						Time
	Layers	**0->0**	**0->1**	**1->0**	**1->1**	**MSE**	**NMSE**	**r**	**%error**	**AIC**	**MDL**	
RBF no GA	0	98.65	1.34	84.4	15.59	0.3305	0.781	0.472	16.78	2100.4	3944	26''
RBF no GA	*1*	*98.65*	*1.34*	*68.8*	*31.19*	*0.300*	*0.710*	*0.538*	*14.13*	*2528.7*	*4690.38*	*31''*
RBF no GA	2	98.99	1.00	87.15	12.84	0.351	0.830	0.412	16.5	2679.7	4866.9	32''
RBF no GA	3	100	0	100	0	0.418	0.989	0.323	20.34	2842.9	5055.7	33''
RBF no GA	4	100	0	100	0	0.442	1.046	0.152	25.59	2922.9	5161.2	33''
RBF no GA	5	100	0	100	0	0.442	1.046	0.06	25.54	2962.3	5226.3	34''
RBF no GA	6	0	100	0	100	0.816	1.930	0.048	46.36	3434.8	5724.3	36''
RBF no GA	7	100	0	100	0	0.466	1.101	0.148	28.44	3079.1	5394	36''
RBF no GA	8	100	0	100	0	0.487	1.151	0.081	30.40	3150.4	5491.1	38''
RBF no GA	9	100	0	100	0	0.472	1.117	-0.032	29.10	3169.1	5535.4	38''
RBF no GA	10	100	0	100	0	0.772	1.826	-0.126	44.82	3555.8	5947.7	39''
RBF, inputs GA	*0*	*97.82*	*2.17*	*33.94*	*66.05*	*0.172*	*0.407*	*0.770*	*10.41*	*-278.7*	*336.8*	*3h42'47''*
RBF, inputs GA	1	96.48	3.51	50.45	49.54	0.25	0.595	0.638	13.10	804.02	1941.7	6h30'56''
RBF, inputs GA	*2*	*97.15*	*2.84*	*22.01*	*77.98*	*0.202*	*0.479*	*0.748*	*14.92*	*691.1*	*1854.5*	*6h29'43''*
RBF, inputs GA	3	100	0	100	0	0.423	1.000	-0.014	19.58	1250.5	2439.5	4h36'56''
RBF, inputs GA	4	100	0	100	0	0.428	1.013	-0.077	22.85	1300.3	2514.8	4h56'42''
RBF, inputs GA	5	100	0	100	0	0.502	1.189	-0.115	31.65	1292.8	2430.5	5h41'42''
RBF, inputs GA	6	100	0	100	0	0.535	1.265	-0.053	33.87	1696.5	3064.6	4h37'14''
RBF, inputs GA	7	0	100	0	100	1.488	3.51	0.163	63.65	2298.8	3590.2	4h56'02''
RBF, inputs GA	8	0	100	0	100	0.864	2.042	0.138	47.93	2434.8	4058.9	5h50'51''
RBF, inputs GA	9	0	100	0	100	0.924	2.185	-0.104	49.81	2042.5	3385.05	5h48'58''
RBF, inputs GA	10	100	0	100	0	0.637	1.507	-0.254	39.36	2460.4	4238.1	7h38'36''
RBF, GA	*0*	*97.31*	*2.68*	*31.19*	*68.80*	*0.177*	*0.418*	*0.763*	*10.676*	*-689.9*	*-349.4*	*5h20'30''*
RBF, GA	1	97.82	2.17	37.61	62.38	0.204	0.484	0.720	11.788	1011.0	2374.0	7h04'30''
RBF, GA	*2*	*96.48*	*3.51*	*23.85*	*76.14*	*0.161*	*0.380*	*0.787*	*8.354*	*-28.709*	*777.56*	*9h34'26''*
RBF, GA	3	100	0	100	0	0.422	0.999	0.349	19.58	1562.5	2951.1	13h22'10''
RBF, GA	4	100	0	100	0	0.420	0.994	0.321	19.65	2860.7	5082.4	20h13'40''
RBF, GA	5	100	0	100	0	0.422	0.999	0.373	19.58	3418.5	5994.7	18h18'46''
RBF, GA	6	100	0	100	0	0.565	1.337	0.426	12.84	4959.8	8391.0	27h43'53''
RBF, GA	7	100	0	100	0	0.422	0.999	0.513	19.58	5038.4	8651.3	36h01'48''
RBF, GA	8	100	0	100	0	0.563	1.333	-0.15	12.72	5861.6	971.3	42h34'22'
RBF, GA	9	100	0	100	0	0.423	1.000	-0.433	19.58	7058.6	11964.1	61h37'32''
RBF, GA	10	100	0	100	0	0.563	1.331	-0.062	35.57	2532.8	4412.8	78h35'57''
RBF, GA, Cross V.	0	98.15	1.84	36.69	63.3	0.174	0.412	0.768	11.096	48.84	869.2	19h36'39''
CV performance		*98.65*	*1.34*	*35.77*	*64.22*	*0.178*	*0.422*	*0.765*	*10.821*	*68.04*	*887.94*	
RBF, GA, Cross V.	1	97.31	2.68	26.60	73.39	0.1163	0.387	0.784	10.12	1980.8	4065.69	17h10'
CV performance		*97.98*	*2.01*	*42.20*	*57.79*	*0.196*	*0.465*	*0.736*	*10.14*	*2112.44*	*4196.09*	
RBF, GA, Cross V.	2	98.65	1.34	37.61	62.38	0.176	0.416	0.769	6.98	4242.09	7741.09	26h14'52''
CV performance		*97.65*	*2.34*	*26.60*	*73.39*	*0.151*	*0.358*	*0.801*	*8.16*	*4139.6*	*7636.65*	
RBF, GA, Cross V.	3	98.15	1.84	26.60	73.39	0.144	0.342	0.812	5.839	2491.49	4958.96	9h39'44''
CV performance		*98.15*	*1.84*	*24.77*	*75.22*	*0.146*	*0.346*	*0.808*	*6.334*	*2503.6*	*4969.7*	
RBF, GA, Cross V.	4	100	0	100	0	0.527	1.247	0.207	10.056	4826.8	8204.0	40h24'46''
CV performance		*100*	*0*	*100*	*0*	*0.518*	*1.224*	*0.261*	*9.263*	*4814.8*	*8190.4*	
RBF, GA, Cross V.	5	100	0	100	0	0.422	0.999	0.433	19.58	4418.3	7634.5	52h20'46''
CV performance		*100*	*0*	*100*	*0*	*0.423*	*0.999*	*0.439*	*19.58*	*4420.04*	*7634.4*	
RBF, GA, Cross V.	6	100	0	100	0	0.422	0.999	0.070	19.58	5292.5	9067.9	63h25'42''
CV performance		*100*	*0*	*100*	*0*	*0.423*	*0.999*	*0.140*	*19.58*	*5294.2*	*9067.5*	
RBF, GA, Cross V.	7	100	0	100	0	0.551	1.304	0.309	11.861	3378.05	5808.41	18h50'12''
CV performance		*100*	*0*	*100*	*0*	*0.551*	*1.302*	*0.284*	*11.803*	*3378.4*	*5807.4*	
RBF, GA, Cross V.	8	100	0	100	0	0.422	0.999	0.385	19.588	7820.1	13213.2	83h53'02''
CV performance		*100*	*0*	*100*	*0*	*0.423*	*0.999*	*0.348*	*19.597*	*7821.9*	*13212.0*	
RBF, GA, Cross V.	9	100	0	100	0	0.422	0.999	0.124	19.58	7540.3	12754.2	89h05'12''
CV performance		*100*	*0*	*100*	*0*	*0.423*	*0.999*	*0.156*	*19.58*	*7541.9*	*12753.0*	
RBF, GA, Cross V.	10	100	0	100	0	0.422	0.999	0.402	19.58	6346.4	10796.3	95h27'19''
CV performance		*100*	*0*	*100*	*0*	*0.423*	*0.999*	*0.391*	*19.58*	*6348.17*	*10795.5*	

Table 2. Results of Radial Basis Function neural networks in WEKA

Neural Net	**0->0**	**0->1**	**1->0**	**1->1**	**Misclass**	**Correct class.**	**MAE**	**MSE**	**RMSE**	**RAE**	**RRSE**	**Time**
RBF	550	21	56	79	77	629	0.1393	0.0748	0.2736	53.12%	78.24%	0.69 sec

(77.9%)	(2.97%)	(7.93%	11.1%	(10.90%)	(89.09%)

V. Conclusions

The RBF neural nets and the hybrid RBF nets indicated the RBF with 3 layers GA in all and CV was the optimal net with an extended processing time of about 9 hours, a quicker solution is given by an RBF with GA on the input layer only with a slightly higher MSE cost in 6 hours. Thirdly the hybrid RBF with GA in all layers results similarly to the RBF with GA and CV with slightly higher MSE and lower r, in the same processing time. The RBF NN was the fastest network but with the highest MSE cost, and the lowest r, as in WEKA data mining platform it had the worst classification results. Thus Genetic Algorithms advance slightly the precision in RBF networks, causing divergence from the initial classifications made by loan experts. On the contrary the regular Radial Basis Function network performed a very fast classification with very low cost achieving a fine fitness to the desired output but with the worst convergence results. Thus the regular Radial Basis Function network with 3 layers and Genetic Algorithms in all the layers and Cross Validation is superior to all the neuro-genetic forms of RBF in Financial Analysis of corporations.

VI. Bibliography

[1] Broomhead D.S. and Lowe D.- Multivariate functional interpolation and adaptive networks. *Complex Systems*, 2:321-355, 1988.

[2] Loukeris N., Matsatsinis N., 'Corporate Financial Evaluation and Bankruptcy Prediction implementing Artificial Intelligence methods', Journal of Business and Economics, Issue 4, Volume 3, April 2006, WSEAS Transactions

[3] NeuroDimensions Inc. With NeuroSolutions software: www.nd.com

[4] Principe J., Euliano N., and Lefebvre W.- Neural and Adaptive Systems: Fundamentals Through Simulations-ed. John Wiley & Sons, Inc., 2000

[5] Tikhonov A.N. and Arsenin V.Y.-*Solutions of Ill-Posed Problems*. Winston, Washington, 1977.

[6] WEKA software: www.cs.waikato.ac.nz/ml/**weka**