# Classification of Large Biomedical Data using ANNs based on BFGS method

I.E. Livieris, M.S. Apostolopoulou, D.G. Sotiropoulos, S.A. Sioutas and P. Pintelas

***Abstract*—Artificial neural networks have been widely used for knowledge extraction from biomedical datasets and constitute an important role in bio-data exploration and analysis. In this work, we proposed a new curvilinear algorithm for training large neural networks which is based on the analysis of the eigenstructure of the memoryless BFGS matrices. The proposed method preserves the strong convergence properties provided by the quasi-Newton direction while simultaneously it exploits the nonconvexity of the error surface through the computation of the negative curvature direction without using any storage and matrix factorization. Moreover, for improving the generalization capability of trained ANNs, we explore the incorporation of several dimensionality reduction techniques as a pre-processing step.**

***Index Terms*—Artificial neural networks, biomedical data, dimensionality reduction, feature extraction, memoryless BFGS, curvilinear search.**

## I. Introduction

During the second half of the last century the areas of biology and medical science have been dramatically changed, from a rather qualitative science that was based on observations of whole organisms to a more quantitative science that is now based on measurements at the molecular level. Nevertheless, the growing research and developments of microarray technology constitute in the exponentially generation of data in size, dimension and complexity. Moreover, these datasets have non-linear relationships between inputs and outcomes, hindering their analysis and modeling. The tremendous amount of data obtained from the microarray studies constitutes a challenge for data analysis which has been focused on developing intelligent computational systems (see [13], [28]), such as artificial neural networks.

Artificial neural networks (ANN) are parallel computational models comprised of densely interconnected, adaptive processing units, characterized by an inherent propensity for learning from experience and also discovering new knowledge. Due to their excellent capability of self-learning and self-adapting, they have been successfully applied in bioinformatics and are often found to be more efficient and accurate than other classification techniques [15]. Mathematically, the problem of training an ANN is highly consistent with the unconstrained optimization theory. More analytically, it can be formulated as the minimization of the error function $E(w)$ defined as the sum of squares of the errors in the outputs [22]. A traditional way to solve this problem is by an iterative gradient-based supervised training algorithm using the update formula

$$w_{k+1} = w_k + \eta_k d_k \tag{1}$$

where $k$ is the current iteration usually called *epoch*, $w_0 \in \mathbb{R}^n$ is a given starting point, $\eta_k > 0$ is the learning rate and $d_k$ is a descent search direction, i.e., $g_k^T d_k < 0$. In the literature, a variety of approaches has been proposed for successfully training large neural networks while most of them use second order information [5], [11]. The most elaborate method is the limited memory BFGS [17], [19] where the search direction in Eq. (1) is defined by building up a Hessian approximation using curvature information from the previous iterations.

In [2] has been proposed a method that exploits the eigenstructure of the memoryless BFGS matrices without using storage and matrix factorization. Consequently, a direction of negative curvature can be computed analytically avoiding the storage and factorization of any matrix. Motivated by their method, we propose a curvilinear scheme which is based on a modification of the memoryless BFGS method for training large neural networks. The proposed algorithm exploits the nonconvexity of the error surface based on information provided by the eigensystem of memoryless BFGS matrices utilizing a pair of directions; a memoryless quasi-Newton direction and a direction of negative curvature, i.e., a direction $d$ such that $d^T \nabla^2 E(w) d < 0$ and it is based on the following iterative form

$$w_{k+1} = \begin{cases} w_k + \eta_k p_k, & \text{if } B_k \text{ is positive definite;} \\ w_k + \eta_k^2 p_k + \eta_k d_k, & \text{otherwise} \end{cases}$$

where $p_k$ is a memoryless quasi-Newton direction, $d_k$ is a direction of negative curvature and $B_k$ is the memoryless BFGS Hessian approximation. In case the Hessian approximation $B_k$ is indefinite the proposed iterative scheme performs a curvilinear search along the path $w_{k+1} = w_k + \eta_k^2 p_k + \eta_k d_k$ which was first introduced by Moré and Sorensen [18]. In different case, the iterative scheme is the standard linesearch procedure (see [3], [20]).

Clearly, the proposed method preserves the strong convergence properties provided by the quasi-Newton direction when $B_k$ is positive definite. Additionally, it exploits the nonconvexity of the error surface through the computation of the negative curvature direction without using any storage and matrix factorization. Moreover, based on the fact that the proposed method uses only inner products and vector summations and requires only $O(n)$ space, it is well-suited for efficiently training large neural networks.

I. E. Livieris is with the Department of Mathematics, University of Patras, Greece. E-mail: livieris@upatras.gr

M. S. Apostolopoulou is with the Department of Mathematics, University of Patras, Greece. E-mail: msa@math.upatras.gr

D. G. Sotiropoulos is with the Department of Informatics, Ionian University, Greece. E-mail: dgs@ionio.gr

S. A. Sioutas is with the Department of Informatics, Ionian University, Greece. Email: sioutas@ionio.gr

P. Pintelas is with the Department of Mathematics University of Patras, Greece. E-mail: pintelas@math.upatras.gr

978-0-7695-3788-7/09 $25.00 © 2009 IEEE
DOI 10.1109/PCI.2009.32

IEEE computer society

Despite, large neural networks can be trained efficiently, these models are usually plagued by poor generalization reliability due to the huge dimension of the dataset. Therefore, to overcome the curse of dimensionality for improving the generalization capability of ANNs the application of a *dimensionality reduction technique* is considered essential, namely the reduction of input dimensionality using a mathematical pre-processing step. More specifically, the goal of dimensionality reduction methods is the transformation of high-dimensional data into a meaningful representation of reduced dimensionality. Hence, with the automatic identification and removal of the less relevant and important inputs we can reduce the size of network and increase its robustness. Therefore, in recent years a variety of nonlinear dimensionality reduction techniques has been proposed in the literature (see [8], [23], [29], [30]) with various properties.

The remainder of this paper is as follows: in Section II we present in details the method to compute the descent directions and describe the proposed algorithm, which is based on the properties of the memoryless BFGS matrices. Section III summarizes traditional dimensionality reduction techniques in order to projecting the original data onto some low dimensional space. This pre-processing step can reduce the size of the ANN classifier which it can now be trained by the classical BFGS method. Simulation results are presented in Section IV and in Section V we give some concluding remarks.

*Notations.* Throughout the paper $\|\cdot\|$ denotes the Euclidean norm and $n$ the dimension of the error function. We indicate that a matrix $A$ is positive definite by $A > 0$ and with $u^{(i)}$ we denote the $i$-th component of vector $u$. The gradient of the error function is denoted by $\nabla E(w_k) = g_k$.

## II. Curvilinear Memoryless BFGS

In this section we briefly discuss the eigenstructure of the Hessian approximation $B$ which is based on the L-BFGS method [17], [19]. The memoryless matrix $B$ is updated by means of the BFGS formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}, \qquad (2)$$

where in the vector pair $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$ is stored curvature information only from the most previous iteration. By setting $B_0 = (1/\theta)I$ in Eq. (2) the resulting minimal memory BFGS update is defined as

$$B_{k+1} = \frac{1}{\theta_{k+1}} I - \frac{s_k s_k^T}{\theta_{k+1} s_k^T s_k} + \frac{y_k y_k^T}{s_k^T y_k}. \qquad (3)$$

Moreover, it is known that the inverse of $B_{k+1}$ is given by the following expression [19]

$$B_{k+1}^{-1} = \theta_{k+1} I - \theta_{k+1} \frac{y_k s_k^T + s_k y_k^T}{s_k^T y_k} + \frac{s_k s_k^T}{s_k^T y_k} \qquad (4)$$

In our approach we consider the case where the scalar parameter $\theta$ is defined as $\theta_{k+1} = \left(s_k^T s_k\right) / \left(s_k^T y_k\right)$ which is the spectral parameter of Barzilai and Borwein [4].

*T*heorem 1 ([2]) Let the symmetric memoryless BFGS matrix defined in (3). Then, the characteristic polynomial of $B_{k+1} \in \mathbb{R}^{n \times n}$ has the general form

$$p(\lambda) = \left(\lambda - \frac{1}{\theta_{k+1}}\right)^{n-2} \left(\lambda^2 - \frac{a_k}{\theta_{k+1}}\lambda + \frac{1}{\theta_{k+1}^2}\right), \qquad (5)$$

where $a_k = 1 + \theta_{k+1} \frac{y_k^T y_k}{s_k^T y_k}$. Moreover, if $a_k > 2$, then $\lambda_1 < 1/\theta_{k+1} < \lambda_n$, where $\lambda_1$ and $\lambda_n$ are the smallest and largest eigenvalues of $B_{k+1}$, respectively.

The parameter $a_k$ is bounded from below by 2, since

$$a_k = 1 + \theta_{k+1} \frac{y_k^T y_k}{s_k^T y_k} = 1 + \frac{\|s_k\|^2 \|y_k\|^2}{(s_k^T y_k)^2} = 1 + \frac{1}{cos^2 \phi} \geq 2,$$

where $\phi$ is the angle between $s_k$ and $y_k$. Clearly, the value of parameter $a_k$ determines if the vectors $s_k$ and $y_k$ are linear independent or not. Hence, the above theorem states that if the vectors are linear independent, that is $a_k > 2$, the extreme eigenvalues are distinct and can be computed by solving the quadratic equation $\lambda^2 - (a_k/\theta_{k+1})\lambda + 1/\theta_{k+1}^2 = 0$. In contrast, if $a_k = 2$, then the characteristic polynomial is reduced to $p(\lambda) = (\lambda - 1/\theta_{k+1})^n$; thus the smallest eigenvalue of $B_{k+1}$ is multiple and equals $\lambda = 1/\theta_{k+1}$. For determining the eigenvector corresponding to the smallest eigenvalue of $B_{k+1}$, we consider the following cases.

In case where the smallest eigenvalue of $B_{k+1}$ is distinct, then the corresponding eigenvector is computed by applying a single step of the inverse iteration. Given a non-zero starting vector $u_0$, inverse iteration generates a sequence of vectors $u_i$, generated recursively by the formula

$$u_i = \left(B - \hat{\lambda} I\right)^{-1} \frac{u_{i-1}}{\|u_{i-1}\|}, \quad i = 1, 2, \ldots$$

where $\hat{\lambda} = \lambda + \epsilon$, $\lambda$ is a distinct eigenvalue of $B$ and $\epsilon \to 0^+$. The sequence of iterates $u_i$ converges to an eigenvector associated with an eigenvalue closest to $\hat{\lambda}$. Moreover, if this particular eigenvalue $\lambda$ is known exactly, this method converges in a single iteration. For being able to apply the inverse iteration, we take into account the following proposition for expressing $(B - \hat{\lambda} I)^{-1}$ in closed form.

*P*roposition 1 ([2]) Let $\Lambda$ be the set of eigenvalues of $B_{k+1}$ with opposite signs. Then, for any $\lambda \in \mathbb{R} \setminus \Lambda$, the matrix $(B_{k+1} + \lambda I)$ is invertible and its inverse can be expressed by the following closed-form

$$(B_{k+1} + \lambda I)^{-1} = \frac{1}{\gamma} \sum_{i=0}^{2} (-1)^i \gamma_i(\lambda) \left(B_{k+1}\right)^i \qquad (6)$$

where the quantities $\gamma = (1/\theta_{k+1} + \lambda)(\lambda^2 + a_k \lambda/\theta_{k+1} + 1/\theta_{k+1}^2)$, $\gamma_2 = 1$, $\gamma_1 = \lambda + (a_k + 1)/\theta_{k+1}$ and $\gamma_0 = \lambda^2 + (a_k + 1)\lambda/\theta_{k+1} + (a_k + 1)/\theta_{k+1}^2$ are functions of $\lambda$.

Hence, using Theorem 1 and Proposition 1 and after some simple algebraic computations, the expression for the eigenvector is defined by $u_1 = \hat{u}_1 / \|\hat{u}_1\|$, where

$$\begin{aligned} \hat{u}_1 &= \sum_{i=0}^{2} (-1)^i \gamma_i(\hat{\lambda}) \left(B_{k+1}\right)^i \frac{u}{\gamma(\hat{\lambda})} \\ &= -\gamma_u(\hat{\lambda})\, u + \gamma_{us}(\hat{\lambda})\, s_k - \gamma_{uy}(\hat{\lambda})\, y_k, \end{aligned} \qquad (7)$$

with $\hat{\lambda} = -\lambda_1 + \epsilon$ , $u = u_0/\|u_0\|$ and the coefficients are

$$\gamma_u(\hat{\lambda}) = \left[1 - \gamma_1(\hat{\lambda})\,\theta_{k+1} + \gamma_0(\hat{\lambda})\,\theta_{k+1}^2\right] / \left[\gamma(\hat{\lambda})\,\theta_{k+1}^2\right],$$

$$\gamma_{us}(\hat{\lambda}) = \left\{\left[1 - \gamma_1(\hat{\lambda})\,\theta_{k+1}\right] s_k^T u + \theta_{k+1} y_k^T u\right\} / \left[\gamma(\hat{\lambda})\,\theta_{k+1}^2 s_k^T s_k\right],$$

$$\gamma_{uy}(\hat{\lambda}) = \left\{\left[1 - \gamma_1(\hat{\lambda})\,\theta_{k+1} + a_k\right] \theta_{k+1} y_k^T u - s_k^T u\right\} / \left[\gamma(\hat{\lambda})\,\theta_{k+1}^2 s_k^T y_k\right].$$

In case where the smallest eigenvalue of $B_{k+1}$ is multiple, then from Theorem 1 we have that $a_k = 2$ and $B_{k+1} = (1/\theta_{k+1})I$. Thus, using the eigendecomposition of $B$ it follows that $B = U \Lambda U^T$, where $U = I$ and $\Lambda = \operatorname{diag}(\lambda_1, \lambda_1, \ldots, \lambda_1)$. It is easy to verify that an eigenvector corresponding to $\lambda_1$ is $u_1 = e_1 = (1, 0, \ldots, 0)^T$.

### A. The *CM-BFGS* training algorithm

At this point, we recall that our new proposed curvilinear scheme uses a pair of directions; a quasi-Newton direction [20] which is defined as

$$p_{k+1} = \begin{cases} -B_{k+1}^{-1} g_{k+1}, & B_{k+1} > 0; \\ -g_{k+1}, & \text{otherwise.} \end{cases} \tag{8}$$

where $B_{k+1}^{-1}$ is defined in equation (4) and a direction of negative curvature [18] which is calculated by

$$d_{k+1} = \begin{cases} 0, & B_{k+1} > 0; \\ -\operatorname{sgn}(u_1^T g_{k+1}) u_1, & \text{otherwise,} \end{cases} \tag{9}$$

where $u_1$ is a normalized eigenvector corresponding to the most negative eigenvalue of $B_{k+1}$. Consequently, we present a high level description of our proposed algorithm based on the Armijo procedure.

```
Step 1: Initiate $w_0$, $0 < c_1 < c_2 < 1$, $Err$ and $\epsilon \to 0$;
    set $k = 0$.
Step 2: If $(E(w_k) < Err)$ or $(\|\nabla E(w_k)\|_2 < \epsilon)$ terminate; else compute the eigenvalues $\lambda_i$ of $B_k$.
Step 3: If $\lambda_1 > 0$ then
    (a) Compute $p_k$; set $d_k = 0$ and $\eta_k = 1$.
    (b) Find $\eta_k > 0$ such that
            $E(w_k + \eta_k p_k) \le E(w_k) + c_1 \eta_k g_k^T p_k$
Step 4: Else if $\lambda_1 \le 0$ then
    (a) Set $p_k = -g_k$ and compute the normalized eigenvector $u_1$; set $d_k = -\operatorname{sgn}\left(u_1^T g_k\right) u_1$ and $\eta_k = 1$.
    (b) Find $\eta_k > 0$ such that
            $E(w_k + \eta_k^2 p_k + \eta_k d_k) \le E(w_k) + c_2 \eta_k \left(g_k^T d_k + \frac{1}{2}\lambda_1\right)$
Step 5: Update the weights
            $w_{k+1} = \begin{cases} w_k + \eta_k p_k, & \text{if } \lambda_1 > 0; \\ w_k + \eta_k^2 p_k + \eta_k d_k, & \text{otherwise} \end{cases}$
Step 6: Compute $g_{k+1}$, $s_k = w_{k+1} - w_k$ and $y_{k+1} = g_{k+1} - g_k$; if $|s_k^T y_k| > 10^{-6}\|s_k\|\|y_k\|$, update the vector pair $\{s_k, y_k\}$.
Step 7: Set $k = k + 1$ and goto Step 2.
```

*Remarks:* In Step 2, the computation of the eigenvalues is based on Theorem 1. In Step 4(a), if $a_k > 2$, then $d_k$ is computed using relation (7), in contrast we set $d_k = -\operatorname{sgn}(g_{k+1}^{(1)})(1, 0, \ldots, 0)^T$. Finally, in Step 6 we skip the update in case $|s_k^T y_k| \le 10^{-6}\|s_k\|\|y_k\|$ to ensure that $B_k$ is well defined.

## III. Dimensionality Reduction

The problem of dimensionality reduction appears in many fields of artificial intelligence such as data mining, data compression and data visualization, moderating the curse of dimensionality and other undesired properties of high dimensional spaces [14]. Given a dataset $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{n \times D}$ consists of $n$ datavectors $x_i$ with dimensionality $D$ and has intrinsic dimension $d$ (with $d << D$). The intrinsic dimensionality of data is the minimum number of parameters needed to account for the observed properties of the data [9]. The goal of dimensionality reduction is the transformation of the dataset $X$ to a new dataset $Y$ with dimensionality $d$ such that certain properties are preserved.

In the literature, there have been proposed several techniques for this problem. Most of them are based on the intuition that data lies on or near a complex low-dimensional manifold that is embedded in the high-dimensional space. These techniques can be summarized in two main groups a) linear techniques (PCA, LPP, OLPP) b) nonlinear techniques (KPCA, LEM, LTSA).

- PCA: Principal component analysis [12] performs a linear mapping of the data to a lower dimensional space in such a way, that the variance of the data in the low-dimensional representation is maximized. The reduction is accomplished by identifying directions, called principal components, along which the variation in the data is maximal.
- LPP: Locality Preserving Projections [10] is a linear dimensional reduction technique which constructs the k-NN graph in order to model the data topology aiming at preserving the local structure defined by the nearest neighbors.
- OLPP: Orthogonal Linear Preserving Projections consists an extension of the LPP algorithm by simply enforcing the mapping to be orthogonal.
- KPCA: Kernel principal component analysis is a nonlinear extension of the traditional PCA that is constructed using a kernel function [25] and it has shown to be a very powerful method of extracting nonlinear features for classification and regression [24].
- LEM: Laplacian Eigenmaps [7] is a dimensionality reduction technique that preserves the local properties of the manifold which are based on the pairwise distances between near neighbors. LEM computes a low-dimensional representation of the data by minimizing a cost function based on the distances between the data points.
- LTSA: Local Tangent Space Analysis [31] is a technique for nonlinear dimensionality reduction that constructs approximations of tangent spaces in order to represent local geometry of the manifold and the global alignment of the tangent spaces to obtain the global coordinate system.

All the above dimensionality reduction techniques have been used in our experimental framework for being able to construct adequate data for training small ANN classifier.

## IV. Experimental Results

We evaluate the generalization performance of our proposed algorithm (`CM-BFGS`) in a variety of biomedical classification benchmarks. Subsequently, we explore the application of a dimensionality reduction technique as a data pre-processing step in the generalization performance of our method. In our experiments, we have selected the following high-dimensional biomedical datasets:

- *Colon Tumor* [D1]: Contains 62 samples collected from colon-cancer patients [1]. Among them, 40 tumor biopsies are from tumors (labeled as ”negative”) and 22 normal (labeled as ”positive”) biopsies are from healthy parts of the colons of the same patients. Two thousand out of around 6500 genes were selected based on the confidence in the measured expression levels.
- *DLBCL-Outcome* [D2] *and DLBCL-Tumor* [D3]: There are two kinds of classifications about diffuse large b-cell lymphoma (DLBCL) addressed in these data [26]. First one is DLBCL versus Follicular Lymphoma (FL) morphology. This set of data contains 58 DLBCL samples and 19 FL samples. The second problem is to predict the patient outcome of DLBCL. Among 58 DLBCL patient samples, 32 of them are from cured patients while 26 of them are from patients with fatal or refractory disease.
- *Lung-Michigan* [D4]: This data set consists of 86 primary lung adenocarcinomas samples and 10 non-neoplastic lung samples are included [6]. Each sample is described by 7129 genes.
- *Central Nervous System-Outcome* [D5]: Patients outcome prediction for central nervous system embryonal tumor [21]. Survivors are patients who are alive after treatment whiles the failures are those who succumbed to their disease. The data set contains 60 patient samples, 21 are survivors and 39 are failures. There are 7129 genes in the dataset.
- *Prostate-Outcome* [D6]: This data set is referred for prediction of clinical outcome [27]. More analytically, 21 patients were evaluable with respect to recurrence following surgery with 8 patients having relapsed and 13 patients having remained relapse free (”non-relapse”) for at least 4 years.

The parameters in `CM-BFGS` were set as $c_1 = c_2 = 10^{-4}$ for all experiments and the initial weights were initiated using the Nguyen-Widrow method. For evaluating classification accuracy of the first five benchmarks we have used the 10-fold cross-validation repeated 100 times while for the last one we have the 4-fold cross-validation. The target dimensionality in all experiments was determined by means of *maximum likelihood* intrinsic dimensionality estimator [16] and for all dimensional reduction techniques we have used the default parameters as in [29]. All simulations have been carried out on a processor Pentium-IV dual core computer (2.0MHz, 1Gbyte RAM) using the neural network toolbox of MATLAB.

Table I presents information about the networks architectures and the total number of weights that were trained on high and low dimensional data for each benchmark. In the right hand of Table I, the number of inputs coincides with the intrinsic dimension $d$ obtained by the maximum likelihood estimator.

|  | High-dimensional data |  |  | Low-dimensional data |  |  |
|---|---|---|---|---|---|---|
| Data Set | Inputs | Neurons in hidden layers | Total weights | Inputs | Neurons in hidden layer | Total weights |
| D1 | 2000 | 10-5 | 20077 | 11 | 6 | 86 |
| D2 | 7129 | 10-5 | 71367 | 22 | 11 | 277 |
| D3 | 7129 | 20-5-10 | 142787 | 24 | 12 | 326 |
| D4 | 7129 | 5-5 | 35692 | 24 | 12 | 326 |
| D5 | 7129 | 20-5-10 | 142787 | 30 | 15 | 497 |
| D6 | 12600 | 10-10 | 126142 | 16 | 8 | 157 |

TABLE I
Neural network architectures

In Table II are summarized the generalization results of ANNs that were trained with `CM-BFGS` algorithm on the high and low dimensional data. Each column reports the average performance in percentage for each dataset using different dimensionality reduction techniques. The column under “None” indicates the results that were obtained using the original data. The best performing technique for a dataset is illustrated in boldface. First of all, we observe that the classification performance of the trained networks was not significantly improved by performing a dimensionality reduction. However, linear techniques significantly outperform nonlinear techniques since they present the best generalization results in five datasets. Additionally, we observe that the traditional PCA is the best reducing technique exhibiting the best overall performance.

| Data | None | PCA | LPP | OLPP | KPCA | LEM | LTSA |
|---|---|---|---|---|---|---|---|
| D1 | 84.5 | **85.4** | 47.6 | 76.9 | 51.5 | 80.1 | 56.8 |
| D2 | 53.7 | 53.7 | **58.1** | 52.0 | 54.8 | 50.9 | 44.3 |
| D3 | 84.0 | **84.5** | 66.0 | 84.3 | 66.4 | 82.1 | 58.9 |
| D4 | 89.5 | 89.6 | 88.7 | **90.1** | 89.2 | 89.4 | 89.4 |
| D5 | 61.3 | 63.9 | 49.1 | 51.2 | **65.0** | 57.3 | 46.0 |
| D6 | 52.3 | 54.7 | 52.8 | **55.6** | 41.9 | 51.8 | 53.6 |

TABLE II
Generalization performance (%) of ANNs trained with CM-BFGS method.

| Data | PCA | LPP | OLPP | KPCA | LEM | LTSA |
|---|---|---|---|---|---|---|
| D1 | **85.4** | 48.5 | 76.7 | 52.9 | 80.6 | 57.3 |
| D2 | 53.9 | **58.9** | 52.7 | 55.3 | 50.1 | 44.0 |
| D3 | **84.1** | 66.0 | **84.1** | 65.9 | 82.9 | 57.2 |
| D4 | 89.6 | 88.9 | **90.0** | 89.2 | 89.5 | 89.4 |
| D5 | 64.5 | 50.3 | 51.0 | **65.8** | 57.9 | 46.0 |
| D6 | 54.4 | 52.8 | **56.1** | 41.8 | 53.4 | 54.2 |

TABLE III
Generalization performance (%) of ANNs trained with the BFGS method.

Table III reports the generalization result of ANNs that were trained with `BFGS` training algorithm (“`trainbfg`”) on the low-dimensional data obtained from the presented dimensionality reduction techniques. Note that the training process is impossible for the BFGS algorithm using the original data. Comparing the results of Table III with the second column ("None") of Table II we observe that both algorithms have similar performance. Therefore, CM-BFGS algorithm is well-suited not only for large-size networks but it can also exhibits satisfactory generalization results on small-size networks.

## V. Conclusions

In this work, we have proposed a neural network classifier for classifying large data obtained from microarrays studies based on a new memoryless BFGS algorithm that incorporates a curvilinear search. The proposed algorithm exploits the nonconvexity of the error surface based on information provided by the eigensystem of memoryless BFGS matrices avoiding any storage and matrix factorization. Furthermore, we have explored the impact of applying a dimensionality reduction technique as a training pre-processing step. Based on our numerical experiments we conclude that the application of linear techniques for dimensionality reduction are capable to improve the generalization ability of our proposed model.

Our future work will be concentrated on extending our framework with other types of classifiers such as support vector machines, decision trees etc in order to gain insight and better analyze the microarray data.

## References

[1] U. Alon, N. Barkai, D.A. Notterman, K. Gishdagger, S. Ybarradagger, D. Mackdagger, and A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 96, pages 6745–6750, 1999.

[2] M.S. Apostolopoulou, D.G. Sotiropoulos, and P. Pintelas. Solving the quadratic trust-region subproblem in a low-memory BFGS framework. *Optimization Methods and Software*, 23(5):651–674, 2008.

[3] L. Armijo. Minimization of functions having Lipschitz continuous partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, 1966.

[4] J. Barzilai and J.M. Borwein. Two point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

[5] R. Battiti. First and second order methods for learning: between steepest descent and Newton's method. *Neural Computation*, 4:141–166, 1992.

[6] D.G. Beer, S.L. Kardia, C.C. Huang, T.J. Giordano, A.M. Levin, D.E. Misek, L. Lin, G. Chen, T.G. Gharib, D.G. Thomas, M.L. Lizyness, R. Kuick, S. Hayasaka, J.M. Taylor, M.D. Iannettoni, M.B. Orringer, and S. Hanash. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature Medicine*, 8(8):816–823, 2002.

[7] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, volume 14, pages 585–591, Cambridge, MA, USA, 2002.

[8] C.J.C. Burges. *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, chapter Geometric Methods for Feature Selection and Dimensional Reduction: A Guided Tour. Kluwer Academic Publishers, 2005.

[9] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press Professional, San Diego, CA, USA, 1990.

[10] X. He and P. Niyogi. Locality preserving projections. *Advances in Neural Information Processing Systems 16 (NIPS 2003)*, 2003.

[11] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Reading, MA, 1991.

[12] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[13] H.T. Huynh, J.J. Kim, and Y. Won. DNA microarray classification with compact single hidden-layer feedforward neural networks. In *FBIT '07: Proceedings of the 2007 Frontiers in the Convergence of Bioscience and Information Technologies*, pages 193–198. IEEE Computer Society, 2007.

[14] L.O. Jimenez and D.A. Landgrebe. Supervised classification in high-dimensional space: geometrical,statistical and asymptotical properties of multivariate data. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 1, pages 39–54, 1997.

[15] B. Lerner, H. Guterman, M. Aladjem, and I. Dinstein. A comparative study of neural network based feature extraction paradigms. *Pattern Recognition Letters*, 20(1):7–14, 1999.

[16] I.S. Lim, P.H. Ciechomski, S. Sarni, and D. Thalmann. Planar arrangement of high-dimensional biomedical data sets by isomap coordinates. In *$16^{th}$ IEEE Symposium on Computer-Based Medical Systems*, pages 50–55, 2003.

[17] D.C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization methods. *Mathematical Programming*, 45:503–528, 1989.

[18] J.J. Moré and D. Sorensen. On the use of directions of negative curvature in a modified Newton method. *Mathematical Programming*, 16:1–20, 1979.

[19] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematical Computing*, 35(151):773–782, 1980.

[20] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.

[21] S.L. Pomeroy, P. P. Tamayo, M. Gaasenbeek, L.M Sturla, M. Angelo, M.E. McLaughlin, J.Y. Kim, L.C. Goumnerova, P.M. Black, C. Lau, J.C. Allen, D. Zagzag, J.M. Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D.N. Louis, J.P. Mesirov, E.S. Lander, and T.R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415:436–442, 2002.

[22] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 318–362, Cambridge, Massachusetts, 1986.

[23] L.K. Saul, K.Q. Weinberger, J.H. Ham, F. Sha, and D.D. Lee. Spectral methods for dimensionality reduction. In *Advances in Neural Information Processing Systems*, volume 17, pages 1473–1480, Cambridge, MA, USA, 2006. MIT Press.

[24] B. Schölkopf, A. Smola, and K.R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[25] J. Shawe-Taylor and N. Christianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, UK, 2004.

[26] M.A. Shipp, K.N. Ross, P. Tamayo, A.P. Weng, J.L. Kutok, R.C. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G.S. Pinkus, T.S. Ray, M.A. Koval, K.W. Last, A. Norton, T.A. Lister, J. Mesirov, D.S. Neuberg, E.S. Lander, J.C. Aster, and T.R. Golub. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(1):68–74, 2002.

[27] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, P. Tamayo, A.A. Renshaw, A.V. D'Amico, J.P. Richie, E.S. Lander, M. Loda, P.W. Kantoff, T.R. Golub, and W.R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, 2002.

[28] C. Soares, L. Montgomery, K. Rouse, and J.E. Gilbert. Automating microarray classification using general regression neural networks. In *Fourth International Conference on Machine Learning and Applications*, pages 508–513. IEEE Computer Society, 2008.

[29] L.J.P. Van der Maaten and H.J. Van den Herik. Dimensionality reduction: A comparative review. *Submitted to Neurocomputing*, 2008.

[30] J. Wang, Z. Zhang, and H. Zha. Adaptive manifold learning. In *Advances in Neural Information Processing Systems*, volume 17, pages 1473–1480, Cambridge, MA, USA, 2005. MIT Press.

[31] Z. Zhang and H. Zha. Principal manifolds and nonlinear dimensionality reduction via local tangent space alignment. *SIAM Journal of Scientific Computing*, 26(1):313–338, 2004.