MINIMUM VARIANCE EXTREME LEARNING MACHINE FOR HUMAN ACTION RECOGNITION

Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki
54124, Thessaloniki, Greece
Email: {aiosif,tefas,pitas}@aiia.csd.auth.gr

ABSTRACT

In this paper we propose an algorithm for Single-hidden Layer Feedforward Neural networks training. Based on the observation that the learning process of such networks can be considered to be a non-linear mapping of the training data to a high-dimensional feature space, followed by a data projection process to a low-dimensional space where classification is performed by a linear classifier, we extend the Extreme Learning Machine (ELM) algorithm in order to exploit the training data dispersion in its optimization process. The proposed Minimum Variance Extreme Learning Machine classifier is evaluated in human action recognition, where we compare its performance with that of other ELM-based classifiers, as well as the kernel Support Vector Machine classifier.

Index Terms— Single-hidden Layer Feedforward Neural networks, Extreme Learning Machine, Human Action Recognition, Classification

1. INTRODUCTION

Extreme Learning Machine is a, relatively, new algorithm for Single-hidden Layer Feedforward Neural (SLFN) netorks training [1] that leads to fast network training requiring low human supervision. Conventional SLFN network training algorithms require the input weights and the hidden layer biases to be adjusted using a parameter optimization approach, like gradient descend. However, gradient descend-based learning techniques are generally slow and may decrease the network's generalization ability, since they may lead to local minima. Unlike the popular thinking that the network's parameters need to be tuned, in ELM the input weights and the hidden layer biases are randomly assigned. The network output weights are, subsequently, analytically calculated. ELM not only tends to reach the smallest training error, but also the smallest norm of output weights. As shown in [2], for feedforward networks reaching a small training error, the smaller the norm of weights are, the better generalization performance the networks tend to have. Despite the fact that the determination of the network hidden layer output is a result of randomly assigned weights, it has been shown that SLFN networks trained by using the ELM algorithm have the properties of global approximators [3]. Due to its effectiveness and its fast learning process, the ELM network has been widely adopted in many classification problems, including human action recognition [4, 5, 6, 7, 8, 9, 10]

Despite its success in many classification problems, the ability of the original ELM algorithm to calculate the output weights is limited due to the fact that the network hidden layer output matrix is, usually, singular. In order to address this issue, the Effective ELM (EELM) algorithm has been proposed in [11], where the strictly diagonally dominant criterion for nonsingular matrices is exploited, in order to choose proper network input weights and bias values. However, the EELM algorithm has been designed only for a special case of SLFN networks employing Gaussian Radial Basis Functions (RBF) for the input layer neurons. In [12], an optimization-based regularized version of the ELM algorithm (ORELM) aiming at both overcoming the full rank assumption for the network hidden layer output matrix and at enhancing the generalization properties of the ELM algorithm has been proposed. ORELM has been evaluated on a large number of classification problems providing very satisfactory classification performance.

By using a sufficiently large number of hidden layer neurons, the ELM classification scheme, when approached from a Discriminant Learning point of view, can be considered as a learning process formed by two processing steps. The first step corresponds to a mapping process of the input space to a high-dimensional feature space preserving some properties of interest for the training data. In the second step, an optimization scheme is employed for the determination of a linear projection of the high-dimensional data to a low-dimensional feature space determined by the network target vectors, where classification is performed by a linear classifier. Based on this observation, the ORELM algorithm has been extended in order to incorporate discriminative criteria in its optimization process [13]. Specifically, it has been shown that the incorporation of the within-class scatter in the optimization process followed for the calculation of the network output weights in-

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

creased the ELM network performance. In this paper, we follow this line of work and propose an extension of the ORELM algorithm which exploits the training data dispersion in the ORELM optimization process. The proposed Minimum Variance ELM (MVELM) algorithm aims at minimizing both the network output weights norm and the dispersion of the training data in the projection space.

We evaluate the proposed MVELM network in human action recognition by employing the Bag-of-Words (BoW)-based video representation and we compare its performance with that of ORELM [12] and MCVELM [13] networks. We also compare the performance of the proposed MVELM network with that of the weighted kernel Support Vector Machine (kSVM) classifier [14, 15], which has been widely used in human action recognition. Finally, we show that the χ^2 kernel [16], which is the state-of-the-art metric for non-linear Bag of Words (BoW)-based action classification can be employed for ELM-based SLFN network training.

The paper is structured as follows. In Section 2 we describe the proposed MVELM algorithm for SLFN network training. Section 3 presents experiments conducted in order to evaluate its performance. Finally, conclusions are drawn in Section 4.

2. THE PROPOSED MINIMUM VARIANCE ELM

The proposed MVELM network is used for supervised classification. Let us denote by $\{\mathbf{x}_i, c_i\},\ i = 1, \ldots, N$ a set of N vectors $\mathbf{x}_i \in \mathbb{R}^D$ followed by class labels $c_i \in \{1, \ldots, C\}$. We would like to employ them in order to train a SLFN network. Such a network consists of D input (equal to the dimensionality of $\mathbf{x}_i$), L hidden and C output (equal to the number of classes involved in the classification problem) neurons. The number of hidden layer neurons is usually selected to be much greater than the number of classes [12, 13], i.e., $L \gg C$.

The network target vectors $\mathbf{t}_i = [t_{i1}, ..., t_{iC}]^T$, each corresponding to a training vector $\mathbf{x}_i$, are set to $t_{ik} = 1$ for vectors belonging to class k, i.e., when $c_i = k$, and to $t_{ik} = -1$ otherwise. Similarly to ELM, the network input weights $\mathbf{W}_{in} \in \mathbb{R}^{D \times L}$ and the hidden layer bias values $\mathbf{b} \in \mathbb{R}^L$ are randomly assigned, while the network output weights $\mathbf{W}_{out} \in \mathbb{R}^{L \times C}$ are analytically calculated. Let us denote by $\mathbf{v}_j$ the j-th column of $\mathbf{W}_{in}$, by $\mathbf{w}_k$ the k-th row of $\mathbf{W}_{out}$ and by w_{kj} the j-th element of $\mathbf{w}_k$. For a given activation function for the network hidden layer $\Phi(\cdot)$ and by using a linear activation function for the network output layer, the output $\mathbf{o}_i = [o_1, \ldots, o_C]^T$ of the network corresponding to $\mathbf{x}_i$ is calculated by:

$$o_{ik} = \sum_{j=1}^{L} w_{kj}\ \Phi(\mathbf{v}_j, b_j, \mathbf{x}_i),\ k = 1, ..., C. \tag{1}$$

It has been shown [12] that, several activation functions $\Phi(\cdot)$ can be used for the calculation of the network hidden layer outputs, like the sigmoid, sine, Gaussian, hard-limiting and Radial Basis Functions (RBF). In our experiments we employ the χ^2 distance function, as will be described in Section 3. By storing the network hidden layer outputs corresponding to the training vectors $\mathbf{x}_i,\ i = 1, \ldots, N$ in a matrix $\mathbf{\Phi}$:

$$\mathbf{\Phi} = \begin{bmatrix} \Phi(\mathbf{v}_1, b_1, \mathbf{x}_1) & \cdots & \Phi(\mathbf{v}_1, b_1, \mathbf{x}_N) \\ \cdots & \ddots & \cdots \\ \Phi(\mathbf{v}_L, b_L, \mathbf{x}_1) & \cdots & \Phi(\mathbf{v}_L, b_L, \mathbf{x}_N) \end{bmatrix}, \tag{2}$$

equation (1) can be expressed in a matrix form as $\mathbf{O} = \mathbf{W}_{out}^T \mathbf{\Phi}$.

By allowing small training errors and trying to minimize both the norm of the network output weights and the variance of the training vectors in the projection space, $\mathbf{W}_{out}$ can be calculated by solving the following optimization problem:

$$\textbf{Minimize:} \quad \mathcal{J} = \|\mathbf{S}_T^{\frac{1}{2}} \mathbf{W}_{out}\|_F^2 + \lambda \sum_{i=1}^{N} \|\boldsymbol{\xi}_i\|_2^2 \tag{3}$$

$$\textbf{Subject to:} \quad \mathbf{W}_{out}^T \boldsymbol{\phi}_i = \mathbf{t}_i - \boldsymbol{\xi}_i,\ \ i = 1, ..., N, \tag{4}$$

where $\boldsymbol{\xi}_i \in \mathbb{R}^C$ is the error vector corresponding to $\mathbf{x}_i$ and λ is a parameter denoting the importance of the training error in the optimization problem. $\boldsymbol{\phi}_i$ is the i-th column of $\mathbf{\Phi}$, i.e., the hidden layer output corresponding $\mathbf{x}_i$. That is, $\boldsymbol{\phi}_i$ is the representation of $\mathbf{x}_i$ in $\mathbb{R}^L$.

In (3), $\mathbf{S}_T$ is a matrix denoting the dispersion of the training vectors in $\mathbb{R}^L$. We have employed the total scatter matrix of the training vectors in $\mathbb{R}^L$ to this end, defined by:

$$\mathbf{S}_T = \sum_{i=1}^{N} (\boldsymbol{\phi}_i - \boldsymbol{\mu})(\boldsymbol{\phi}_i - \boldsymbol{\mu})^T, \tag{5}$$

where $\boldsymbol{\mu} \in \mathbb{R}^L$ is the mean vector of the entire training set in $\mathbb{R}^L$, i.e., $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\phi}_i$.

By analyzing (3) we obtain:

$$\begin{aligned} \mathcal{J} &= \|\mathbf{S}_T^{\frac{1}{2}} \mathbf{W}_{out}\|_F^2 + \lambda \sum_{i=1}^{N} \|\boldsymbol{\xi}_i\|_2^2 \\ &= Tr\left(\mathbf{W}_{out}^T \mathbf{S}_T \mathbf{W}_{out} \right) + \lambda \sum_{i=1}^{N} \|\mathbf{W}_{out}^T \boldsymbol{\phi}_i - \mathbf{t}_i\|_2^2 \\ &= \sum_{i=1}^{N} \left(\left(\mathbf{W}_{out}^T \boldsymbol{\phi}_i - \mathbf{W}_{out}^T \boldsymbol{\mu} \right)^T \left(\mathbf{W}_{out}^T \boldsymbol{\phi}_i - \mathbf{W}_{out}^T \boldsymbol{\mu} \right) \right) \\ &+ \lambda \sum_{i=1}^{N} \|\mathbf{W}_{out}^T \boldsymbol{\phi}_i - \mathbf{t}_i\|_2^2 \\ &= \sum_{i=1}^{N} \left(\|\mathbf{o}_i - \mathbf{o}\|_2^2 + \lambda \|\mathbf{o}_i - \mathbf{t}_i\|_2^2 \right), \end{aligned} \tag{6}$$

where $\mathbf{o} = \mathbf{W}_{out}^T \boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{W}_{out}^T \boldsymbol{\phi}_i$ is the mean network output of the entire training set. That is, the minimization of

$\mathcal{J}$ leads to the determination of the network output weights $\mathbf{W}_{out}$ which provide a compromise between the training vectors dispersion in the network output space $\mathbb{R}^C$ and the network training error, according to the regularization parameter λ.

By substituting (4) in $\mathcal{J}$ (3) and solving for $\frac{\vartheta \mathcal{J}}{\vartheta \mathbf{W}_{out}} = 0$, $\mathbf{W}_{out}$ is given by:

$$\mathbf{W}_{out} = \left(\mathbf{\Phi}\mathbf{\Phi}^T + \frac{1}{\lambda}\mathbf{S}_T \right)^{-1} \mathbf{\Phi}\mathbf{T}^T. \quad (7)$$

It should be noted here that the calculation of the training vectors dispersion in the network hidden layer space $\mathbb{R}^L$, rather than in the input space $\mathbb{R}^D$, has the advantage that nonlinear relationships between the training vectors $\mathbf{x}_i$ can be exploited.

After the determination of the network output weights $\mathbf{W}_{out}$, a test vector $\mathbf{x}_t$ can be introduced to the trained network and be classified to the class corresponding to the maximal network output:

$$c_t = arg \max_k o_{tk}, \;\; k = 1, \ldots, C. \quad (8)$$

3. EXPERIMENTS

In this section we present experiments conducted in order to evaluate the performance of the proposed MVELM network. We have employed two publicly available action recognition datasets, namely the Hollywood2 and the Olympic Sports datasets. As a baseline approach we use the state-of-the-art method proposed in [15, 17]: we employ the BoW-based video representation by using HOG, HOF and MBH descriptors evaluated on the trajectories of densely sampled interest points. Classification is performed by employing a kernel SVM classifier and the χ^2 kernel [16].

For the ELM-based classification schemes, we have also employed the χ^2 distance function:

$$\Phi(\mathbf{x}_i, \mathbf{v}_j) = exp\left(\frac{1}{2A} \sum_{d=1}^{D} \frac{(x_{id} - v_{jd})^2}{x_{id} + v_{jd}} \right), \quad (9)$$

where A is a parameter used to scale the χ^2 distance between the training vector $\mathbf{x}_i$ and network input weight vector $\mathbf{v}_j$. We set the value of A equal to the mean χ^2 distance between the network hidden weights. Here it should be noted that, since χ^2 distance is used for a histogram-based data representation, the input weight vectors should be formed by positive values and should be normalized in order to have unit l_1 norm.

In the case of ELM-based classification, multiple descriptors are combined by using the network output sum [18, 19]:

$$\mathbf{o}_t = \sum_n \mathbf{o}_t^n, \quad (10)$$

where $\mathbf{o}_i^n$ is the network output for descriptor n.

Regarding the optimal value of the regularization parameter λ used in all the ELM-based classification schemes, as well as in the kSVM classifier, it has been determined by following a linear search strategy. That is, for each classifier, multiple experiments have been performed by employing different parameter values ($\lambda = 10^r$, $r = -6, \ldots, 6$) and the best performance is reported. The number of hidden layer neurons has been set equal to $L = 1000$ for all the ELM-based classification schemes, which has been shown to provide satisfactory performance in many classification problems [12].

In the following, we describe the datasets and evaluation measures used in our experiments. Experimental results are provided in subsection 3.3.

3.1. The Hollywood2 dataset

The Hollywood2 dataset [20] consists of 1707 videos collected from 69 Hollywood movies. The actions appearing in the dataset are: answering the phone, driving car, eating, ghting, getting out of car, hand shaking, hugging, kissing, running, sitting down, sitting up, and standing up. Example video frames of the dataset are illustrated in Figure 1. A standard training-evaluation split is provided in the dataset (823 videos are used for training and performance is measured in the remaining 884 videos). Training and test videos come from different movies. The performance is evaluated by computing the average precision (AP) for each action class and reporting the mean AP over all classes (mAP), as suggested in [20].

Fig. 1. Video frames of the Hollywood2 dataset depicting instances of all the twelve actions.

3.2. The Olympic Sports dataset

The Olympic Sports dataset [21] consists of 783 videos which have been collected from YouTube and annotated using Amazon Mechanical Turk. The actions appearing in the dataset are: high-jump, long-jump, triple-jump, pole-vault, basketball lay-up, bowling, tennis-serve, platform, discus, hammer, javelin, shot-put, springboard, snatch, clean-jerk and vault.

	Traj	HOG	HOF	MBHx	MBHy	Combined
kSVM [15]	60.5 %	63 %	58.7 %	67.4 %	67.7 %	74.1 %
ORELM [12]	59.64 %	66.45 %	58.09 %	67.1 %	69.51 %	74.86 %
MCVELM [13]	60.56 %	**66.84**%	**58.74**%	68.87 %	71.06 %	76.74 %
MVELM	**60.83**%	66.51 %	57.85 %	**71.12**%	**73.68**%	**78.61**%

Table 1. Performance (mAP) on the Olympic Sports dataset.

	Traj	HOG	HOF	MBHx	MBHy	Combined
kSVM [15]	47.8 %	41.2 %	**50.3**%	**48.5**%	**52.9**%	58.2 %
ORELM [12]	46.95 %	40.95 %	48.68 %	45.98 %	50.21 %	57.65 %
MCVELM [13]	47.32 %	**42.03**%	49.45 %	47.8 %	51.33 %	58.34 %
MVELM	**48.12**%	41.61 %	49.55 %	46.91 %	51.52 %	**58.93**%

Table 2. Performance (mAP) on the Hollywood2 dataset.

Example video frames of the dataset are illustrated in Figure 2. A standard training-evaluation split is provided in the dataset (649 videos are used for training and performance is measured in the remaining 134 videos). The performance is evaluated by computing the mean Average Precision (mAP) over all classes, as suggested in [21].

Fig. 2. Video frames of the Olympic Sports dataset.

3.3. Experimental Results

Tables 1, 2 illustrate the mean average precision values obtained by applying the four classification schemes on the Olympic Sports and the Hollywood2 datasets, respectively. As can be seen in these Tables, the performance of the ELM-based classification schemes is comparable with that of the kernel SVM classifier. Specifically, it can be seen that in the Olympic Sports dataset, the ORELM network outperforms kSVM in three, out of six, cases. The proposed MVELM algorithm, as well as the MCVELM algorithm, increase the performance of the ELM network. The MCVELM network outperforms all the other classification schemes in two cases, while the proposed MVELM network provides the highest performance in four cases. Overall, the proposed MVELM network provides the best performance, equal to 78.61% by using a combination of the five descriptors.

In the Hollywood2 dataset, kSVM clearly outperforms the ORELM in all the cases. Similar to the Olympic Sports case, the MCVELM and the proposed MVELM algorithms increase the performance of the ELM network. It can be seen that the proposed MVELM network outperforms the kSVM classifier in three, out of six, cases. Overall, the proposed MVELM network provides the best performance, equal to 58.93%, by using a combination of the five descriptors.

4. CONCLUSIONS

In this paper we proposed an algorithm for Single-hidden Layer Feedforward Neural networks training. The proposed algorithm extends the Extreme Learning Machine algorithm in order to exploit the dispersion of the training data, represented in the network hidden layer space, in its optimization process. The performance of the proposed Minimum Variance Extreme Learning Machine algorithm has been evaluated in human action recognition by employing the BoW-based video representation and the χ^2 kernel function, where it has been found to outperform other ELM-based classification schemes and the kernel SVM classifier.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART).

5. REFERENCES

[1] G.B. Huang, Q.Y. Zhu, and C.K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," *IEEE International Joint Conference on Neural Networks*, 2004.

[2] P.L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 525–536, 1998.

[3] G.B. Huang, L. Chen, and C.K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[4] A. Iosifidis, A. Tefas, and I. Pitas, "Dynamic action recognition based on dynemes and extreme learning machine," *Pattern Recognition Letters*, vol. 34, pp. 1890–1898, 2013.

[5] H.J. Rong, G.B. Huang, and Y.S. Ong, "Extreme learning machine for multi-categories classification applications," *IEEE International Joint Conference on Neural Networks*, 2008.

[6] Y. Lan, Y.C. Soh, and G.B. Huang, "Extreme learning machine based bacterial protein subcellular localization prediction," *IEEE International Joint Conference on Neural Networks*, 2008.

[7] T. Helmy and Z. Rasheed, "Multi-category bioinformatics dataset classification using extreme learning machine," *IEEE Evolutionary Computation*, 2009.

[8] A. Iosifidis, A. Tefas, and I. Pitas, "Multi-view human action recognition under occlusion based on fuzzy distances and neural networks," *European Signal Processing Conference*, 2012.

[9] A.. Iosifidis, A. Tefas, and A. Pitas, "Active classification for human action recognition," *IEEE International Conference on Image Processing*, 2013.

[10] A.. Iosifidis, A. Tefas, and A. Pitas, "Dynamic action classification based on iterative data selection and feedforward neural networks," *European Signal Processing Conference*, 2013.

[11] Y. Wang, F. Cao, and Y. Yuan, "A study on effectiveness of extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2483–2490, 2011.

[12] G.B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[13] A. Iosifidis, A. Tefas, and I. Pitas, "Minimum class variance extreme learning machine for human action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, D.O.I. 10.1109/TCSVT.2013.2269774.

[14] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," *British Machine Vision Conference*, 2009.

[15] H. Wang, A. Klaser, C. Schmid, and C.L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 60, pp. 1–20, 2013.

[16] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International Journal of Computer Vision*, vol. 73, pp. 213–238, 2007.

[17] H. Wang, A. Klaser, C. Schmid, and C.L. Liu, "Action recognition by dense trajectories," *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[18] J. Kittler, R. Hatef, and J. Duin, R. and. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[19] A. Iosifidis, A. Tefas, and I. Pitas, "View-invariant action recognition based on artificial neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 412–424, 2012.

[20] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[21] J.C. Niebles, C.W. Chend, and L. Fei-Fei, "Modeling temporal structure of decomposable mition segemnts for activity classification," *European Conference on Computer Vision*, 2010.