Optimum Learning of Artificial Neuron

Kateřina Nováková [1], *Jaromír Kukal* [2]

[1] CTU Prague, Faculty of Nuclear Sciences and Physical Engineering
[2] ICT Prague, Department of Computing and Control Engineering

Abstract: Learnable artificial neuron is a basic element of ANN. There are various approaches to its learning. The techniques of constrained minimization were used instead of traditional back-propagation. The learning tasks were converted to several optimization tasks and then solved using optimization toolbox in Matlab. Numerical results were compared and discussed.

Keywords: neural network, perceptron, pattern recognition, optimization, Matlab.

1 Introduction

Our aim was to prove the possibility of using the perceptron network to classify binary images, whereas the weights were adjusted through constrained optimization. We can use various methods to recognize 2D objects, but many of them depend on translation, scaling, or rotation of the image. One of the methods, independent of these transformations, is called moment invariants method. The TSR invariant system based on invariant moments was introduced by Hu [3]. Preprocessing includes binary object separation; calculation of general, central, standardized, and invariant moments; and standardization of invariant moments. The traditional perceptron was used for final recognition.

2 Task description

Having a set of 2D binary images of various types, shifts, sizes, and angles of rotation, we tried to categorize them by means of perceptron network. We used ten classes of 2D objects. Every class is represented by sixteen objects in the training set (TRS) and using two objects in the testing (verification) set (TSS).

3 Invariant preprocessing

Any general binary image can be represented as a function $\mathrm{f} : \mathbb{R}^2 \rightarrow \{0; 1\}$. Three types of moments are defined in the literature [2]: general, central, standardized central moments. On the basis of these moments Hu [3] developed the system of seven TSR invariant moments:

$$
\begin{aligned}
\varphi_1 &= \nu_{20} + \nu_{02} \\
\varphi_2 &= (\nu_{20} - \nu_{02})^2 + 4\nu_{11}^2 \\
\varphi_3 &= (\nu_{30} - 3\nu_{12})^2 + (3\nu_{21} - \nu_{03})^2 \\
\varphi_4 &= (\nu_{30} + \nu_{12})^2 + (\nu_{21} + \nu_{03})^2 \\
\varphi_5 &= (\nu_{30} - 3\nu_{12}) \cdot (\nu_{30} + \nu_{12}) \cdot [(\nu_{30} + \nu_{12})^2 - 3(\nu_{21} + \nu_{03})^2] + \\
&+ (3\nu_{21} - \nu_{03}) \cdot (\nu_{21} + \nu_{03}) \cdot [3(\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2] \\
\varphi_6 &= (\nu_{20} - \nu_{02}) \cdot [(\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2] + 4\nu_{11}(\nu_{30} + \nu_{12})(\nu_{21} + \nu_{03}) \\
\varphi_7 &= (3\nu_{21} - \nu_{03}) \cdot (\nu_{30} + \nu_{12}) \cdot [(\nu_{30} + \nu_{12})^2 - 3(\nu_{21} + \nu_{03})^2] - \\
&- (\nu_{30} - 3\nu_{12}) \cdot (\nu_{21} + \nu_{03}) \cdot [3(\nu_{30} + \nu_{12})^2 - (\nu_{21} + \nu_{03})^2],
\end{aligned}
$$

where ν_{pq} is standardized central moment of $p + q$ order.

4 Invariant pattern set

Every 2D binary object is than converted to the vector $\vec{\varphi} = (\varphi_1, \ldots, \varphi_7) \in \mathbb{R}^7$ which represents the object in 7^{th} dimensional vector space. Aplying the previous principle to $m \in \mathbb{N}$ objects, we obtained the matrix representation of given pattern set

$$\mathbf{\Phi} = \begin{bmatrix} \Phi_{11}, & \dots, & \Phi_{17} \\ \vdots & & \\ \Phi_{m1}, & \dots, & \Phi_{m7} \end{bmatrix},$$

where Φ_{ij} is the value of j^{th} invariant for the i^{th} object, $m \in \mathbb{N}$ is the number of objects. With regard of unbalanced intervals of components, the column standardization is necessary. Then the standardized matrix $\Phi^{\oplus}$ has the following form

$$\Phi_{ij}^{\oplus} = \frac{\Phi_{ij} - E(\Phi_{kj})}{S(\Phi_{kj})} \sim N(0,1),$$

where $k = 1, \dots, m$, m is the number of objects. The task will be solvable if the data measured are separable. Owing to linear nonseparability of one class in 7D, we must approach feature space extension. The new vector of features will have the form $\vec{\psi} = (\psi_1, \dots, \psi_7, \psi_1^2, \psi_1 \cdot \psi_2, \dots, \psi_1 \cdot \psi_7, \psi_2^2, \psi_2 \cdot \psi_3, \dots, \psi_7^2) \in \mathbb{R}^{35}$ where $\psi_1, \dots, \psi_7$ are standardized components of the original vector $\vec{\varphi} = (\varphi_1, \dots, \varphi_7)$. Experiment proved that all the data are linearly separable in this 35^{th} dimensional space.

5 Nonlinear SSQ minimization

It is possible to use

- Newton method
- Gauss - Newton method
- Levenberg - Marquardt method

for nonlinear SSQ minimization [4].

As regards the Newton method convergence, it is not so good and this method has a number of drawbacks. The Gauss - Newton method solves the criterion minimization problem in the form of least squares

$$\mathrm{f}(\vec{x}) = \frac{1}{2}||\mathbf{h}(\vec{x})||^2 = \frac{1}{2}\mathbf{h}^T(\vec{x})\mathbf{h}(\vec{x}),$$

where $\mathbf{h}(\vec{x})$ is a nonlinear function. Hence, they are nonlinear least squares. The plain form of the Gauss - Newton method is based on the linearization of function $\mathbf{h}(\vec{x})$ at point $\vec{x}_k$. Then

$$\mathbf{h}(\vec{x}) \doteq \mathbf{h}(\vec{x}_k) + \bigtriangledown\mathbf{h}(\vec{x}_k)(\vec{x} - \vec{x}_k).$$

We get the new iteration by means of the minimization of the linear approximation norm, and hence

$$\vec{x}_{k+1} = \vec{x}_k - (\bigtriangledown\mathbf{h}^T(\vec{x}_k) \bigtriangledown \mathbf{h}(\vec{x}_k))^{-1}(\bigtriangledown\mathbf{h}(\vec{x}_k))^T\mathbf{h}(\vec{x}_k).$$

Provided matrix $(\bigtriangledown\mathbf{h}^T(\vec{x}_k) \bigtriangledown \mathbf{h}(\vec{x}_k))$ is singular, we make it regular by choosing such diagonal matrix $\triangle_k$ in order that matrix $(\bigtriangledown\mathbf{h}^T(\vec{x}_k) \bigtriangledown \mathbf{h}(\vec{x}_k) + \triangle_k)$ should be positively definite. Choosing matrix $\triangle_k$ as $\triangle_k = \alpha_k \mathrm{I}, (\alpha_k > 0)$, we get the so-called Levenberg - Marquardt method. Then the iterative algorithm is

$$\vec{x}_{k+1} = \vec{x}_k - (\bigtriangledown\mathbf{h}^T(\vec{x}_k) \bigtriangledown \mathbf{h}(\vec{x}_k) + \alpha_k \mathrm{I})^{-1}(\bigtriangledown\mathbf{h}(\vec{x}_k))^T\mathbf{h}(\vec{x}_k).$$

While coefficient α_k is small, the method approximates the Gauss - Newton method, whereas when coefficient α_k approaches infinity, we approximate the steepest descent mehod. The problem is to choose the appropriate coefficient α_k. First it must be as big as possible to advance towards the optimum, and when we are near the optimum, then, on the contrary, it must be small to warrant the algorithm convergence.

6 Perceptron

This neural network represents a model of the nonlinear system. Perceptron [1] is a neural network with n input and one output neuron. The projection of input signal $(x_1, \dots, x_n)$ to the output is given by the formula

$$y = \mathrm{f}\,(w_0 + \sum_{k=1}^{n} w_k x_k),$$

where it is necessary to determine weights w_k for $k = 0, \ldots, n$. Weights determination leads to solving the problem of nonlinear least squares. This problem is called learning or training of the neural network. We can take advantage of the perceptron neural network for classification of m input vectors arranged to a matrix. Let

$$y_i^* = \mathrm{f}\left(x_{i1}, \ldots, x_{in}, w_0, \ldots, w_n\right), \quad \text{for } i = 1, \ldots, m,$$

where

$$\mathrm{f}(\vec{x}_i, \vec{w}) = \tanh\left(\sum_{k=0}^{n} w_k x_{ik}\right), \ x_{i0} = 1,$$

and y_i^* is the required i^{th} input vector response. Then the sum of nonlinear least squares LSQ is expressed by the formula

$$LSQ = \sum_{i=1}^{m} \left(\tanh\left(\sum_{k=0}^{n} w_k x_{ik}\right) - y_i^*\right)^2.$$

In our case, we have a neural network with 35 input neurons accordant with components of features vector from extended feature space and ten output neurons accordant with particular object types (classes). We put to the input matrix Ψ with m rows corresponding to all objects, and $n = 35$ columns corresponding to vector $\vec{\psi}$ components.

7 Experimental part

7.1 Constrained least squares

Statement of the problem to resolve

$$LSQ = \min,$$

on condition that

$$||\vec{w}||_p \leq R,$$

where

$$LSQ = \sum_{i=1}^{m} \left(\tanh\left(\sum_{k=0}^{n} w_k x_{ik}\right) - y_i^*\right)^2,$$

$$R = \text{const}.$$

A special Matlab function was created for resolving this issue. It provides Levenberg-Marquardt method for finding weights while the above condition is required. The algorithm was tested with parameters $p = 2, R = 40$. On training data there was only one mistake, no mistake on testing data occurred.

7.2 Minimization of p-norm with nonlinear constraint

Statement of the problem to resolve

$$||\vec{w}||_p = \min$$

on condition that

$$LSQ \leq LSQ^*,$$

where

$$LSQ = \sum_{i=1}^{m} \left(\tanh\left(\sum_{k=0}^{n} w_k x_{ik}\right) - y_i^*\right)^2,$$

$$LSQ^* = \text{const}.$$

The standard Matlab tool `fmincon` function and some additional functions were used for resolving this task. Additional functions provide the p-norm and LSQ computation. The algorithm was tested with parameters $p = 2, LSQ^* = 10$. Three mistakes on the training data occurred, no mistake on the testing data was found.

7.3 Minimization of p-norm with linear constraint

Statement of the problem to resolve

$$||\vec{w}||_p = \min$$

on condition that

$$\left|\tanh\left(\sum_{k=0}^{n} w_k x_{ik}\right) - y_i^*\right| \le \delta,$$

$$\delta = \text{const.}$$

Constraint linearization provides

$$\left|\tanh\left(\sum_{k=0}^{n} w_k x_{ik}\right) - y_i^*\right| \le \delta$$

$$-\delta \le \tanh\left(\sum_{k=0}^{n} w_k x_{ik}\right) - y_i^* \le \delta$$

$$-\delta + y_i^* \le \tanh\left(\sum_{k=0}^{n} w_k x_{ik}\right) \le \delta + y_i^*$$

$$\text{atanh}(-\delta + y_i^*) \le \sum_{k=0}^{n} w_k x_{ik} \le \text{atanh}(\delta + y_i^*)$$

Thus

$$\sum_{k=0}^{n} w_k x_{ik} \le \text{atanh}(\delta + y_i^*),$$

$$-\sum_{k=0}^{n} w_k x_{ik} \le -\text{atanh}(-\delta + y_i^*)$$

just when

$$-1 \le \delta + y_i^* \le 1,$$

$$-1 \le -\delta + y_i^* \le 1.$$

Otherwise the adequate constraint is eliminated.

A new function for linearization process was created and standard matlab tool `fmincon` was used for optimization with linear constraints. The algorithm was tested with parameters $p = 2, \delta = 0.1$ No mistake on the trainig and the testing data occurred.

7.4 Classifier comparison

Three previous approaches were compared on real pattern set. Several quality measures were used for the comparison:

- **Ep** – average mean value of positive patterns
- **Sp** – average standard deviation of positive patterns
- **En** – average mean value of negative patterns
- **Sn** – average standard deviation of negative patterns
- **TRSe** – number of mistakes in training set
- **TSSe** – number of mistakes in testing set.

The numeric results are collected in Tab. 1.

Classifier	Ep	Sp	En	Sn	TRSe	TSSe
Constrained least squares	0.89	0.06	− 0.98	0.05	1	0
Minimization of p-norm with nonlinear constraint	0.43	0.11	− 0.87	0.06	3	0
Minimization of p-norm with linear constraint	0.96	0.01	− 0.96	0.01	0	0

Table 1: Classifier comparison

8 Conclusion

The numeric experiments prove the possibility of using perceptron neural network for classifying binary objects transformed by translation, scaling, and rotation. For object description seven features based on moments were used, whereas projection of all extended 35^{th} dimensional vectors to the plain by PCA proved their separability. The adjustment of feature space was made by the process of standardization. Vectors from the extended feature space were used as input for perceptron. All methods are acceptable, but the best results were achieved using minimization of p-norm with a linear constraint. Classifiers are not noise resistant. Already a small measure of noise is of the cardinal importance for classification. A possible improvement is given by using the median filter, however, it is not included in this work.

References

[1] Fausett L.: Fundamentals of Neural Networks. Prentice – Hall, New Jersey, 1994

[2] Hlaváč V., Šonka M.: Počítačové vidění. Grada, Praha, 1992

[3] Hu M. K.: Pattern Recognition by Moment Invariance. In: Proceedings of the Institute of Radio Engineers, 49, 1428; 1961

[4] Štecha J.: Optimální rozhodování a řízení. Vydavatelství ČVUT, Praha, 2000

Contacts

Kateřina Nováková
Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering
Trojanova 13, 120 00 Prague 2, Czech Republic
E-mail: katka.novakova@dc.fjfi.cvut.cz

Jaromír Kukal
Prague Institute of Chemical Technology
Faculty of Chemical Engineering
Department of Computing and Control Engineering
Technická 1905, 166 28 Prague 6, Czech Republic
Phone: +420-2-24354212, fax: +420-2-24355053
E-mail: jaromir.kukal@vscht.cz