Pattern Recognition 77 (2018) 204–215

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Training neural network classifiers through Bayes risk minimization applying unidimensional Parzen windows

Marcelino Lázaro [a,*], Monson H. Hayes [b], Aníbal R. Figueiras-Vidal [a]

[a] Signal Theory and Communications Department Universidad Carlos III de Madrid, Spain
[b] Department of Electrical and Computer Enginering of George Mason University, Fairfax, VA, USA

ARTICLE INFO

Article history:
Received 15 June 2017
Revised 26 October 2017
Accepted 18 December 2017
Available online 21 December 2017

Keywords:
Bayes risk
Parzen windows
Binary classification

ABSTRACT

A new training algorithm for neural networks in binary classification problems is presented. It is based on the minimization of an estimate of the Bayes risk by using Parzen windows applied to the final one-dimensional nonlinear transformation of the samples to estimate the probability of classification error. This leads to a very general approach to error minimization and training, where the risk that is to be minimized is defined in terms of integrated one-dimensional Parzen windows, and the gradient descent algorithm used to minimize this risk is a function of the window that is used. By relaxing the constraints that are typically applied to Parzen windows when used for probability density function estimation, for example by allowing them to be non-symmetric or possibly infinite in duration, an entirely new set of training algorithms emerge. In particular, different Parzen windows lead to different cost functions, and some interesting relationships with classical training methods are discovered. Experiments with synthetic and real benchmark datasets show that with the appropriate choice of window, fitted to the specific problem, it is possible to improve the performance of neural network classifiers over those that are trained using classical methods.

1. Introduction

Rosenblatt proposed the Perceptron Rule to train a two-class linear discriminant in the late 1950s [1,2]. It can be considered as the first Learning Machine (LM). In the field of Statistics, soft activations - called link functions - appeared as the result of considering different classes of likelihoods or probabilistic solutions (see [3], Ch. 4). This is the case of logistic and probit regressions, that use a sigmoid or a Gaussian distribution as activation functions for the linear combinations. Although several works presented the chain rule approach to train multi-layer networks [4–6] that include other activations to build non-linear transformations of the input samples, it was not until the Back-Propagation (BP) algorithm was introduced in 1986 [7,8] that Multi-Layer Perceptrons (MLPs) received a great deal of attention and found many practical applications, including ensemble forms [9–11] to increase their expressive capabilities. The appearance of Support Vector Machines [12,13] that employ the kernel trick [14,15] and impose a hinge cost diminished the interest in MLPs in the late 1990s and early 2000s, but the introduction of Deep Learning (DL) architectures and algorithms [16–18] put them again in the focus of current research.

Bayesian formulations [19] play a central role in analytical studies of decision and classification [20,21]. However, for the important class of discriminative (non-generative) LM classifiers, the only well-studied connection with Bayesian theory is to get estimates of the "a posteriori" probabilities of the hypotheses at the output of a LM trained by means of Bregman divergences [22]. Overviews of this subject from the perspective of Machine Learning may be found in [23–25].

In this paper, we will establish a general and direct correspondence between Bayesian risk minimization and LM classifier training for binary classification, via modeling the one-dimensional output of the neural network by means of the Parzen windows method [26] to estimate probability densities. Addressing just binary cases is not a serious limitation, because binarizing multi-class problems provides better (ensemble) machine designs than using classical soft-max forms [10,27]. Using single machines for multi-class problems would impose a multi-threshold decision, which is difficult to design and will degrade performance, or it will require multi-dimensional kernels, creating serious difficulties in their design. The direct connection between the windows that are applied in the Parzen estimator and the cost or risk function that is minimized emerges immediately, showing that several well known cost functions are particular cases of the general framework

* Corresponding author.
E-mail address: mlazaro@tsc.uc3m.es (M. Lázaro).

https://doi.org/10.1016/j.patcog.2017.12.018
0031-3203/

that is proposed. Some experiments show that this perspective can help to improve the performance of classical LM classifiers.

We want to emphasize that this approach merges discriminative training with generative approaches concepts: The overall training process is carried out according to discriminative principles, but the last step consists of modeling the non-linear transformation from input patterns to the output of the network by means of unidimensional Parzen windows. This permits to combine some advantages of both families of techniques, such as high performance and robustness against imbalanced situations. To avoid any kind of confusion, we want to strongly remark that we are proposing to apply Parzen windowing just at the output level of the LM classifier, in the definition of the cost to be minimized during training. This definition is independent of the classifier architecture, whose internal layers can use any form of activations, even including radial basis functions.

The rest of the paper is organized as follows. The basic formulation of our approach and the gradient-type algorithms that serve to optimize the estimated Bayesian objective are introduced in Section 2. Some characteristics of Parzen windows, focusing on their role in the training algorithm, are discussed in Section 3. In Section 4, the learning rules obtained using some particular windows are presented, and the equivalence of some of them with classical methods such as the perceptron rule is proved. A series of experiments that illustrate the benefits of adopting some forms of windows are presented in Section 5. The main conclusions of this work and some avenues for further research close our contribution.

2. Training of neural network classifiers minimizing Bayesian risk by Parzen windows

The basic problem of binary classification may be described simply as follows. Given a training set $\mathcal{T}$,

$$\mathcal{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\} \tag{1}$$

consisting of N pairs of labeled patterns, $(\mathbf{x}_i,\ y_i)$, where $\mathbf{x}_i$ are vectors and $y \in \{\pm 1\}$ are target values that represent one of two classes, it is assumed that there is some unknown target function,

$$f : \mathcal{X} \longrightarrow \mathcal{Y} \tag{2}$$

mapping $\mathbf{x}$ to y that is to be learned from the training data. The goal is then to find a function $g(\mathbf{x})$ within a set of functions, $\mathcal{F}$, for predicting y from $\mathbf{x}$, where the function minimizes some error, or is optimal according to some criterion. In this work, the set of functions (classifiers) that are considered are those that correspond to a neural network with a single output and a threshold-based decision. Thus, each function in $\mathcal{F}$ is the soft output of the network that is a nonlinear function,

$$z = g(\mathbf{x}, \mathbf{w}) \tag{3}$$

where $\mathbf{w}$ is a set of trainable parameters, and the decision rule of the classifier is

$$\widehat{y} = \mathrm{sgn}(z) \tag{4}$$

The analytical expression of $g(\mathbf{x}, \mathbf{w})$ in terms of the parameters $\mathbf{w}$ depends on the architecture of the neural network. The proposed training method is valid for every possible architecture, such as an MLP with one or several hidden layers, or a Radial Basis Function (RBF) network, just to mention the most common architectures; and with every possible activation function in the neurons of the network (hyperbolic tangent, rectified linear units, Gaussian units for RBF's, etc.). But it can also be applied to a linear classifier, i.e., $z = g(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\mathbf{x}$.

Once that the neural network architecture is fixed, and thus the analytical expression of $g(\mathbf{x}, \mathbf{w})$ is fixed, the neural network parameters that are to be found are those that minimize the following simplified Bayes' risk[1]

$$\mathcal{R} = c_{-1}\Pr(\widehat{y}=1|y=-1)\Pr(y=-1) + c_1\Pr(\widehat{y}=-1|y=1)\Pr(y=1) \tag{5}$$

where c_i is the cost of making an error when the correct class is i. The probabilities $\Pr\{y=i\}$ may be estimated from the relative number of samples of each class in the training set, but estimating the conditional probabilities can be more difficult. However, since $\mathbf{x}$ is classified according to the decision rule $\widehat{y} = \mathrm{sgn}(z)$, where the output of the neural network, z, is one-dimensional, then the conditional probabilities are

$$\Pr(\widehat{y} = 1|y=-1) = \int_0^\infty p(z|y=-1)\ dz \tag{6}$$

$$\Pr(\widehat{y} = -1|y=1) = \int_{-\infty}^0 p(z|y=1)\ dz \tag{7}$$

Because the conditional densities $p(z|y)$ are unknown, a large number of training samples in each class may be necessary in order to estimate them accurately. However, z is a one-dimensional variable, and all that is required are estimates of the integrals of those conditional densities, and not the densities themselves, so it may not be as critical to have a large training set. Therefore, we consider to use Parzen window estimates of the conditional densities $p(z|y=i)$ from the set of outputs $\{z_n\}$ associated to the labeled training set $\{(\mathbf{x}_n,\ y_n)\}$ to obtain an estimate of the Bayes risk (5). Note that this approach is notably different of using Parzen windows to obtain estimates of the conditional distributions of the input, $p(\mathbf{x}|y=i)$ or the joint input-output distributions, $p(\mathbf{x},\ y)$, such as in [28,29]. These distributions related with the input patterns are multi-dimensional, while here Parzen method is applied to estimate conditional densities at the output of the neural network, $p(z|y=i)$, which are one-dimensional. Parzen window estimates of these distributions are as follows

$$\widehat{p}(z|y=i) = \frac{1}{N_i}\sum_{n\in S_i} k_i(z-z_n) \quad ; \quad i \in \{\pm 1\} \tag{8}$$

where

$$S_1 = \{n : y_n = 1\} \quad \text{and} \quad S_{-1} = \{n : y_n = -1\} \tag{9}$$

and where N_i is the number of samples in S_i and $k_i(z)$ is the Parzen window used to estimate $p(z|y=i)$. Note that, in order to be a valid window, it is necessary that $k_i(z) \geq 0$ and has unit area. Substituting the Parzen estimate of the conditional densities into the conditional probabilities in Bayes' risk gives

$$\Pr(\widehat{y} = 1|y=-1) = \frac{1}{N_{-1}}\sum_{n\in S_{-1}}\int_0^\infty k_{-1}(z-z_n)dz \tag{10}$$

$$\Pr(\widehat{y} = -1|y=1) = \frac{1}{N_1}\sum_{n\in S_1}\int_{-\infty}^0 k_1(z-z_n)dz \tag{11}$$

Since these probabilities involve integrals of the Parzen windows, define $K_i(z)$ to be the integral of the window,

$$K_i(z) = \int_{-\infty}^z k_i(\alpha)d\alpha \tag{12}$$

An example is given in Fig. 1, where the Parzen window is a rectangular pulse. Note that since $k_i(z)$ has the form of a probability

[1] Note that in this definition of the risk, the costs of taking correct decisions in the classical Bayesian formulation have been neglected, which is a common assumption that does not limit the validity of the formulation.